1972

Digital registration of stereo images using the fast Fourier transform

Victor Francis Wilreker Jr.

DIGITAL REGISTRATION OF STEREO IMAGES USING THE FAST FOURIER TRANSFORM

by

VICTOR FRANCIS WILREKER, JR., 1943-

A THESIS

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI - ROLLA

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

1972

Approved by

William H. Tranter (Advisor)

ABSTRACT

The exact alignment of multispectral imagery is a requirement of any image classification system. This investigation develops a system of programs which result in a technique that corrects for the translational and rotational misalignment of two images. Rotational and translational components are determined by using the fast Fourier transform to find the points of maximum correlation between the two images. The registration of two equal but misaligned boxes is given as an example of how the transform technique is used in correlating the images. The results of registering two digitized images that were obtained by scanning a pair of color and color infrared transparencies is included.

ACKNOWLEDGEMENTS

The author wishes to thank Dr. W. H. Tranter for his valuable assistance and guidance in completing this investigation. He also wishes to thank the National Science Foundation, RANN (Research Applied to National Needs) Lead Study Program, for its past and continued support of the "New Lead Belt" project, the McDonnell Aircraft Company, St. Louis, Missouri for scanning the stereo pictures, the Missouri Geological Survey, Rolla, Missouri for taking the pictures, and the Computer Center staff at the University of Missouri - Rolla Campus for their assistance.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd) PAGE

LIST OF FIGURES

LIST OF MAPS

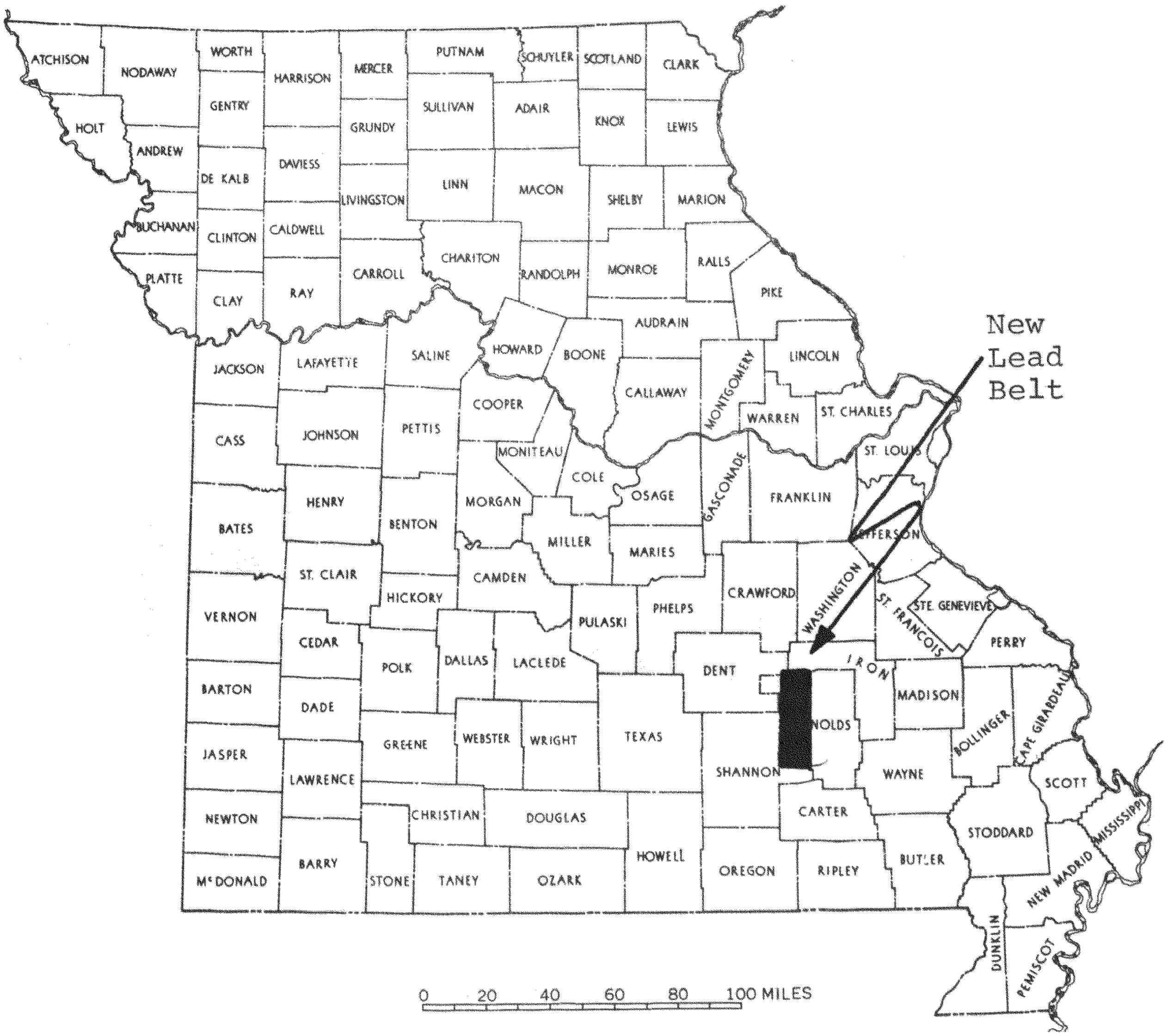
LIST OF TABLES

I. INTRODUCTION

The new lead belt of southeastern Missouri is recognized as one of the world's largest discovered lead deposits, and an extensive mining operation is now underway. This has resulted in the release of large amounts of lead, copper, zinc, cadmium, and other heavy metals into a formerly unaffected ecosystem. Consequently, this area, Map I, represents an ideal site in which to study the effects of these materials on a wide variety of biogeochemical systems and to develop techniques to evaluate and control the effects.

Recognizing the uniqueness of the area, the University of Missouri - Rolla is carrying out an extensive investigation to determine the environmental effects of this industrial development. This investigation is being conducted with the support of the National Science Foundation, RANN (Research Applied to National Needs) Lead Study Program, concerned industries, and several State and Federal agencies.[1]

It was recognized early in the project that remote sensing is an important tool in the study of environmental pollution and should play a role in the lead belt data gathering program.[2] The study area covers approximately 400 square miles and extensive sampling of the entire area is impractical because of the time necessary to collect and process the samples and the expense of analyzing a large



Map I

The new lead belt of southeastern Missouri (Permission for use granted by the Missouri Geological Survey) number of samples. Remote sensing using aerial photography allows data to be collected for very large areas inexpensively and with relative ease. The data obtained by remote sensing techniques can be correlated with the ground truth which has been collected by other members of the research team. In this way their investigations can easily be expanded to large geographical areas. All meaningful remote sensing programs must be accompanied by these ground truth investigations if consistant results are to be expected.[3]

Considerable accuracy can often be realized from this type of remote sensing, especially if multispectral data collecting and processing techniques are utilized. This can be accomplished in two ways. The most efficient method is to use a multispectral line scanner in which the surface being sensed is linearly scanned. The scan, which is wideband reflected radiation from the target, is decomposed into multiple channels. Each channel contains information corresponding to the radiation falling into a predetermined spectral window. This information is converted into an analog voltage which can be digitized. With this type of sensor, an image of the area scanned does not exist until the data is processed. The technique requires expensive equipment for its application but has the advantage of allowing many channels of data to be collected simultaneously over very wide bandwidths. Additionally, the bandwidth of individual scanner channels can be quite narrow.[4]

An alternate technique is to photograph the area in question using several cameras with film sensitive in several different bandwidths. The resulting photographs can then be scanned with a flying-spot scanner and the scans can be digitized. This technique has several disadvantages. The overall bandwidth of the multispectral line scanner cannot usually be achieved because of the difficulty of obtaining film sensitive in the required spectral ranges. Also, the narrow bandwidths obtainable on individual channels of a line scanner are usually not realizable with photography. However, multispectral photography is inexpensive and simple to obtain. Thus, it has found widespread application in environmental research[5] and is worthy of additional investigation.

One inherent difficulty with multispectral photography is that an image registration problem usually exists if multiple images are to be examined simultaneously. For this case, multispectral photographs must be registered to insure that corresponding samples from each photograph yield information relevant to corresponding samples on the surface under investigation.[6]

It is this registration problem that is our primary concern. The purpose of this investigation is to develop and implement a system for image registration. The system developed operates in three phases. First, we perform multiple two dimensional correlations of two images using the fast Fourier transform, then we determine the displacement vectors to the points of maximum correlation in each two dimensional correlation. This yields the displacement vectors required to register the two images.

II. REVIEW OF THE LITERATURE

Multispectral remote sensing using both line scanning and photography has gained an excellent reputation as a data collection tool for many types of problems of national interest. The ERTS-A (Earth Resources Technology Satellite) vehicle carries both a multispectral line scanner and a four channel return beam vidicon system for gathering multispectral data. This satellite was very recently launched (July 1972), but the data obtained is already proving valuable in many study areas. These include land use mapping and land use change detection,[7] geological and geophysical remote sensing,[8] evaluation of agriculture resources,[9] and water resource management.[10]

Prior to the ERTS investigation, there were many remote sensing programs which utilized multispectral techniques. Perhaps the best known program has been carried out by the LARS (Laboratory for Applications of Remote Sensing) Laboratory at Purdue University in Lafayette, Indiana. The major objective of the LARS investigations was the automated classification of crop and soil types.[11] Purdue has also played an important role in the detection of diseased corn. This study proved important in combating the corn blight which has infected the south and midwest in recent years.[12] The University of Tennessee has also participated in an effort aimed at the detection of damaged agriculture using multispectral techniques.[13]

The two dimensional correlation coefficient is a useful tool in solving the registration problems that arise when various types of multispectral remote sensing systems are implemented.[6,14] Development of the fast Fourier transform algorithm[15] has enhanced these systems by greatly reducing the number of machine operations required to find the correlation coefficients.

III. SCANNING AND DIGITIZING IMAGES

Before we investigate the registration of multispectral images, a word is in order concerning the method used to convert the images obtained from aerial photography to digital arrays which are suitable inputs to digital image processing systems.

Photographs can be digitized by scanning either positive or negative images with a flying spot scanner. The scanner measures the density or transmittance of the image at each picture element. This value is then digitized using an analog to digital converter. This process results in a matrix, wherein each entry of the matrix is a digital word which represents the picture element corresponding to that matrix position. Since the scanner measures the density or transmittance of each picture element, the digital word represents only a gray level. The number of gray levels which can be used for each picture element is determined by the number of quantization levels established by the analog to digital converter. For binary digital systems the number of quantization levels used for picture processing is typically 2^n, where n is an integer typically in the range $1 \leq n \leq 8$.

Figures 3-1 and 3-2 represent digitized color and color infrared images obtained from a low-flying aircraft in a region of the lead belt. The images were obtained by scanning color and color infrared transparencies with the flying

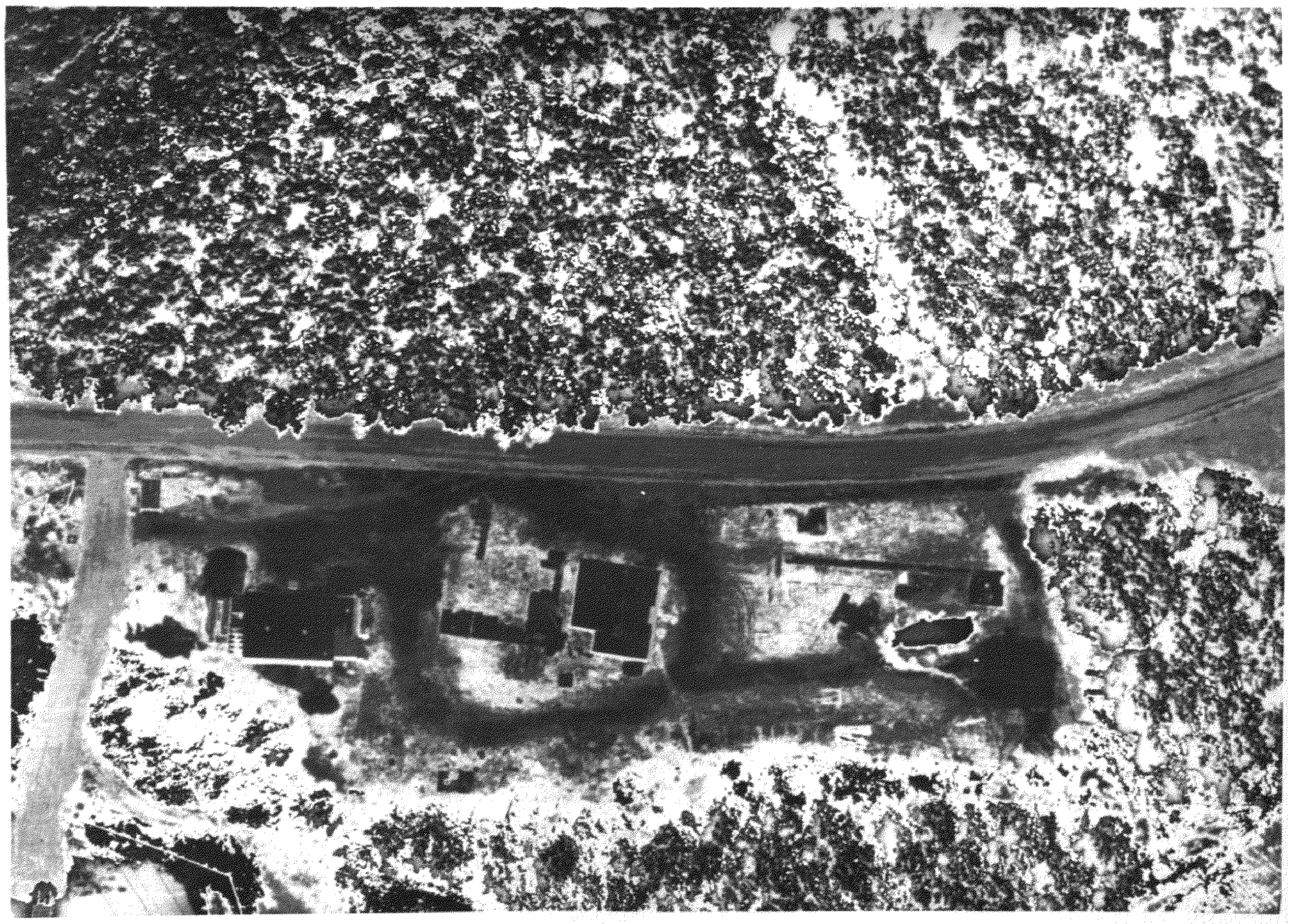


Figure 3-1 Digitized Color Image

Figure 3-2 Digitized Color Infrared Image

spot scanner located at the Reconnaissance Laboratory of McDonnell Aircraft Company in St. Louis, Missouri. The specifications of the scanner are shown in Table I.

Figure 3-3 shows the histograms which represent the distribution of the quantization levels of each image. The differences between the color and color infrared histograms are easily seen. For example, the peak at level 4 of the color infrared image does not appear in the color image. This peak suggests the existance of a feature observable in the color infrared image but not observable in the color image.

Simple image processing can often be accomplished using only one digitized image and its histogram. For example, the third major peak in the histogram of the color image corresponds roughly to the road and similar areas illustrated in Figure 3-1. This is shown by printing all picture elements having quantization levels between 38 and 46 as Ø.
This operation yields the processed image shown in Figure 3-4.

It can be seen from Figure 3-4 that definition of the road is relatively poor. This results because of the poor classification accuracy usually obtainable with single images. The classification accuracy can be improved by examining multiple picture elements from multiple photographs obtained in different spectral ranges with a multi-spectral camera. However, the proper utilization of

Table I

Specifications of Scanning and Digitizing System

Scan Area	4.5" x 4.5" (max.)
Spot Size	Recording Spot-0.0005" (min.)
	Recording Spot-0.0007" (min.)
Scanner Resolution	56 line pairs/mm (cutoff)
Scanning Speed	3 in/sec (max.)
Video Encoded	Density and Transmittance
Sample Density	4096 samples/line (max.)
Word Length	6 bits plus parity (corresponds to 64 gray or quantization levels)
Digital Recording	IBM 7-track format
	200 bits/inch packing density

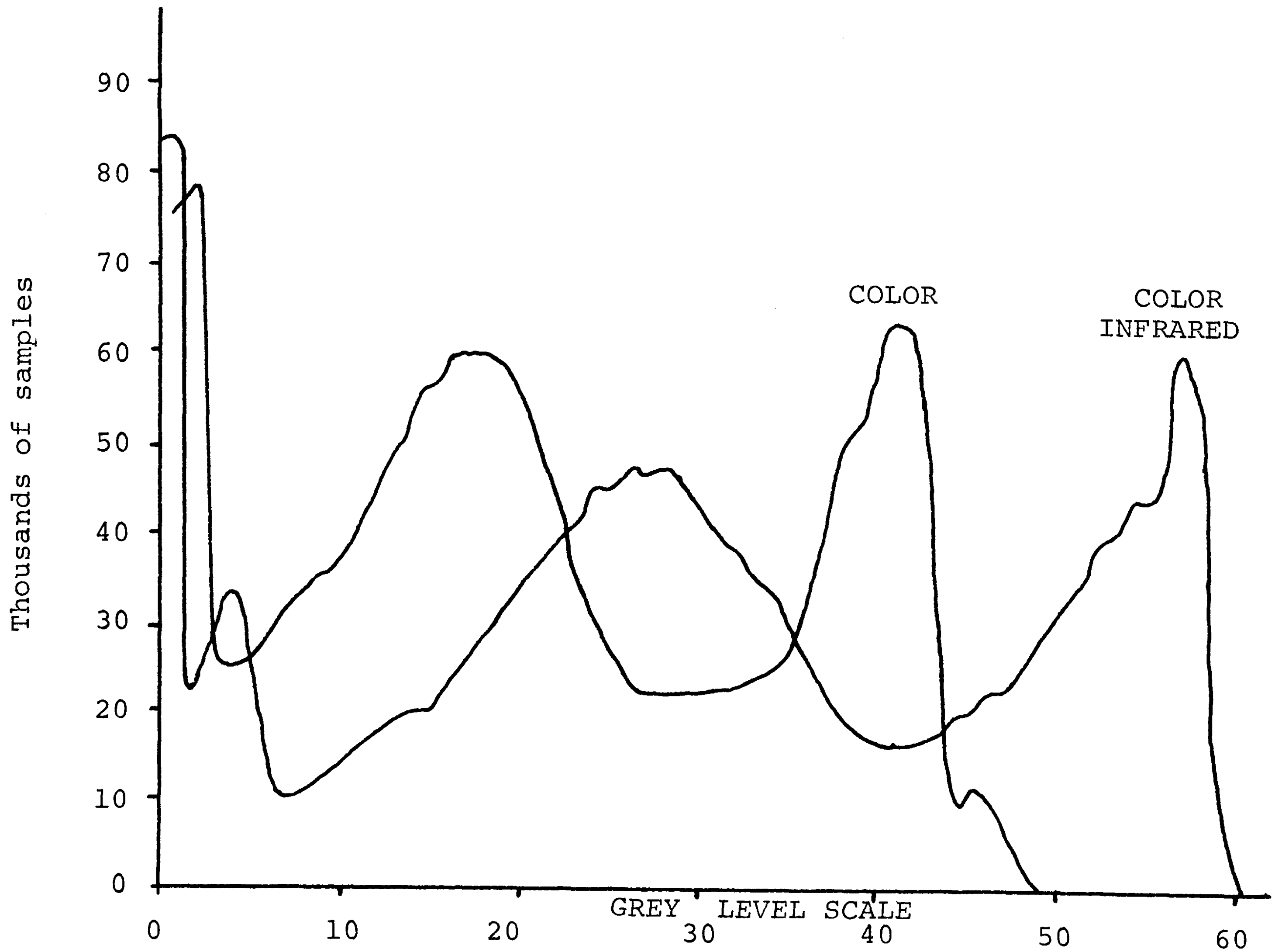


Figure 3-3 Histograms of Color and Color Infrared Image Scans

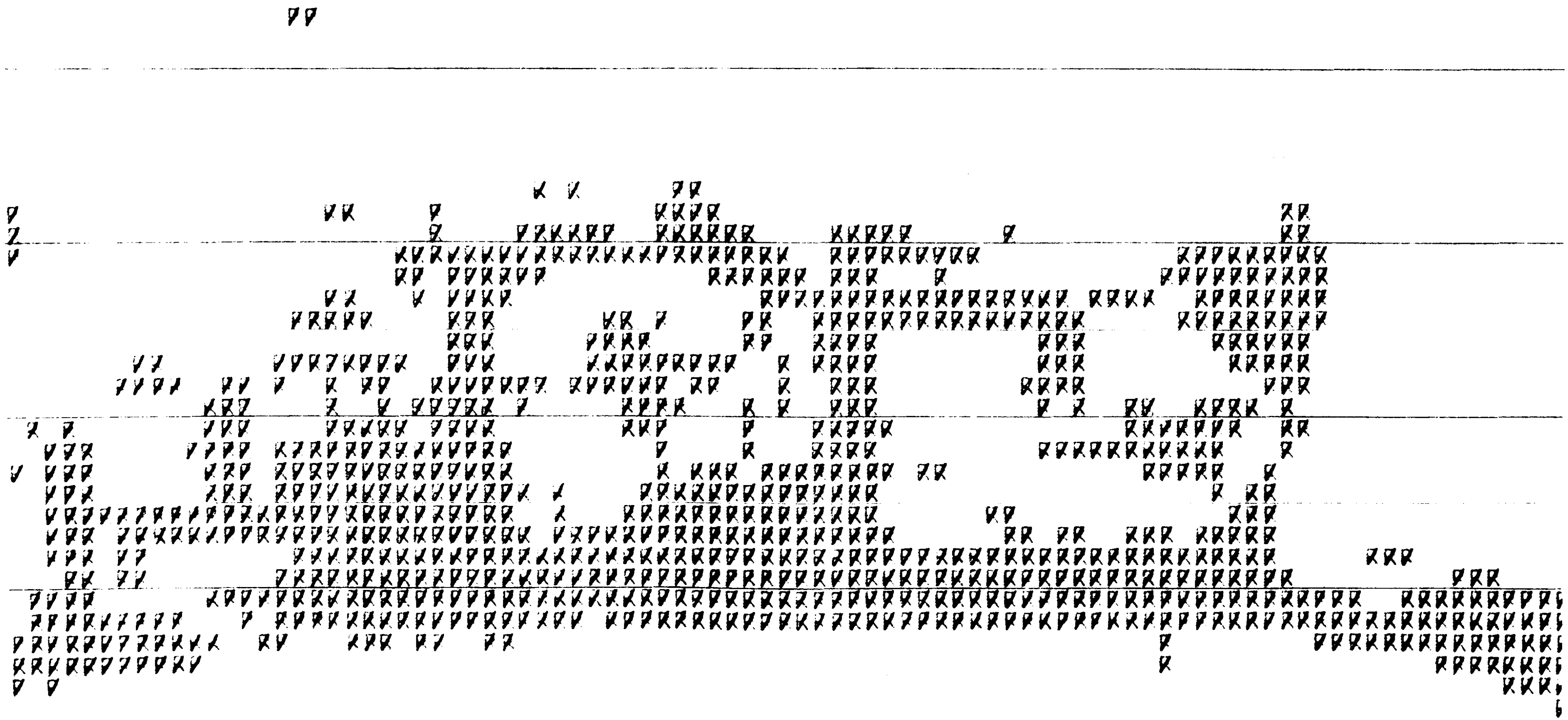


Figure 3-4 Classification of Road Using the Digitized Color Image, Figure 3-1, and Its Grey Scale Histogram, Figure 3-3

multiple picture elements assumes perfect registration.

For perfect registration, the same physical location must be addressable within both images. For example, suppose the corner of a building is addressed in one image, that same corner must be addressable in the other image if classification is to take place. If the second image is actually addressing the lawn adjacent to the corner of the building, the two images are misaligned, and correct classification cannot take place.

Misalignment of images can occur for three reasons: rotation, translation, and scale differences. Since this investigation is only concerned with multispectral photographs taken at the same time and altitude with similar cameras having high quality lenses, the scale aberrations will not be a subject of this study. Registering the two images will consist of finding the rotational and translational components of misalignment and then operating on one of the images to correct and align the two images.

Of the image registration techniques available, correlation gives one of the best estimates of the spatial displacement. Correlation is especially suitable because of its inherent normalization and smoothing. However, correlation also requires many two dimensional numerical integrations in the pictorial (image) domain. In computer processing, the time required for these integrations becomes prohibitive.

There is another method for obtaining the correlation coefficient without having to perform the multiple integrations. It is well-known that correlation of signals in the time domain is equivalent to multiplication in the Fourier or frequency domain.[16] This also applies to correlation of images. Correlation in the pictorial domain is equivalent to multiplication in the spatial domain.

Although correlation via the transform method requires less machine calculation than the normal correlation calculation, the transform method still requires three numerical integrations, and was not extensively used for large arrays until the development of the fast Fourier transform (FFT). The FFT calculates the Fourier transform by using the periodic nature of the transform to reduce the number of machine operations from M^2 to $M \log_2 (M)$, where the base image over which the correlation takes place is an M by M picture element array.

IV. THEORETICAL CONSIDERATION

As stated previously, the two dimensional correlation function will be used to determine the translational and rotational components needed to align the two images. This section will be concerned with how the elements of the components are found using the correlation function. Also to be shown is how the Fourier transform is used to find the correlation function. An example of digital correlation will be presented to illustrate the theory.

The continuous two dimensional correlation function, c(x,y) is defined as

$$c(x,y) = \frac{\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(\alpha,\beta)\; B(\alpha+x,\beta+y)\; d\alpha d\beta - \eta_a \eta_b}{[V_a V_b]^{\frac{1}{2}}} \qquad (4.1)$$

where A and B are the two images being correlated and x and y are the shifts of B with respect to A along the X and Y coordinate axis. V_a and V_b are the variances of the respective A and B images, and η_a and η_b are the means of A and B.

There is an alternate method for calculating the correlation function which makes use of the Fourier transform. Correlation of two images in the pictorial domain is the convolution of one image with the conjugate of the other image in the spatial domain.

Assume the two images have zero mean and unity variance, then

$$c'(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(\alpha,\beta)\ B(\alpha+x,\beta+y)\ d\alpha d\beta. \qquad (4.2)$$

Writing B by its Fourier transform, we have

$$c'(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(\alpha,\beta)\ [\frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H(\tau,\lambda) \cdot$$

$$\text{Exp}[j2\pi\{(x+\alpha)\tau+(y+\beta)\lambda\}]d\tau d\lambda]d\alpha d\beta \qquad (4.3)$$

where

$$H(\tau,\lambda) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} B(\alpha,\beta) \cdot$$

$$\text{Exp}\{-j2\pi[(x+\alpha)\tau+(y+\beta)\lambda]\}\ d\alpha d\beta. \qquad (4.4)$$

Interchanging the order of integration results in

$$c'(x,y) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H(\tau,\lambda)\ \text{Exp}[j2\pi(x\tau+y\lambda)] \cdot$$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(\alpha,\beta)\ \text{Exp}[j2\pi(\alpha\tau+\beta\lambda)]\ d\alpha d\beta]\ d\tau d\lambda \qquad (4.5)$$

from which it is easily shown that the middle integral is the complex conjugate of the Fourier transform of A

$$G^*(\tau,\lambda) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A(\alpha,\beta)\ \text{Exp}[j2\pi(\alpha\tau+\beta\lambda)]\ d\alpha d\beta. \qquad (4.6)$$

Therefore,

$$c'(x,y) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H(\tau,\lambda)G^*(\tau,\lambda)\ \mathrm{Exp}[j2\pi(x\tau+y\lambda)]\ d\tau d\lambda \tag{4.7}$$

or

$$c'(x,y) = F^{-1}[H(\tau,\lambda)G^*(\tau,\lambda)] \tag{4.8}$$

where F^{-1} is the inverse Fourier transform operator, $H(\tau,\lambda)$ is the Fourier transform of the B image and $G(\tau,\lambda)$ is the complex conjugate of the A image. We see that the correlation function can be found by independently calculating the Fourier transform of both images and then inverse transforming their product.

If A and B consist of discrete picture elements, as would be the case in a digital image processing system, the correlation function (4.2) becomes an array of coefficients, where each element in the array is

$$c'(x_i,y_i) = \sum_{\alpha=1}^{N} \sum_{\beta=1}^{N} A(\alpha,\beta)B(\alpha+x_i,\beta+y_i) \tag{4.9}$$

where

$$-x_o \le x_i \le x_o \qquad \text{and} \qquad -y_o \le y_i \le y_o$$

and A and B are sampled images of size N by N. The result is that the B image (correlating image) is being moved with respect to the A image after each pair of numerical integrations.

To make all the correlation coefficients unbiased, the A image (base image) must be larger than B. It is then possible for B to move with respect to A and not overlap the edge of A. If B does overlap the edge of A, the resulting correlation coefficient would be biased and therefore of no use. Since the B array is to be shifted $2x_o + 1$ increments along the X coordinate axis and $2y_o + 1$ increments along the Y coordinate axis (the plus one is the zero shift calculation), the total size of the base array must be $2x_o + N$ by $2y_o + N$. If the number of shifts in each coordinate direction is equal ($2x_o = 2y_o = S$), then the base image becomes an M by M array, where $M = N + S$.

To make the numerical integrations uniform, the correlating array is centered in an M by M array of zeros. This allows the correlating array to move with respect to the base image and still not have the actual N by N image go off the edge of the base image. Equation (4.8) becomes

$$c'(x_i, y_i) = \sum_{\alpha=1}^{M} \sum_{\beta=1}^{M} A(\alpha,\beta) B(\alpha+x_i, \beta+x_i) \qquad (4.10)$$

$$-S \leq x_i \leq S \qquad \text{and} \qquad -S \leq y_i \leq S.$$

Performing a two dimensional correlation in this manner requires $(2S + 1)^2 M^2$ additions and multiplications plus the mean and variance calculations. For typical size arrays ($M = 20$ to $M = 100$), and shifts ($S = 10$ to $S = 60$), it is easily seen that the number of machine operations quickly becomes prohibitive.

The straight Fourier transform method of obtaining the correlation coefficient does not reduce the number of calculations to any great extent. The transform technique has been enhanced by the development of the fast Fourier transform,[15] which reduces the number of operations required to produce the transform from M^2 to $M \log_2(M)$. Therefore, the correlation coefficients can be calculated using the fast Fourier transform by having to perform $3 M \log_2(M) + M^2$ operations plus the mean and variance calculations.

As an example of how correlation is accomplished using the fast Fourier transform, the two boxes in Figures 4-1 and 4-2 will be correlated. Both boxes have amplitude of one and a width of fourteen. Box number one is centered in a 64 by 64 array of zeros, while box number two has been offset by four rows and four columns.

The magnitude of the Fourier transform of both boxes is shown in Figure 4-3. Interpretation of the figure is simplified if the figure is divided into four quadrants centered at (33, 33). Quadrant 1 is located at the right of the figure. Quadrant 2 is in the background of the

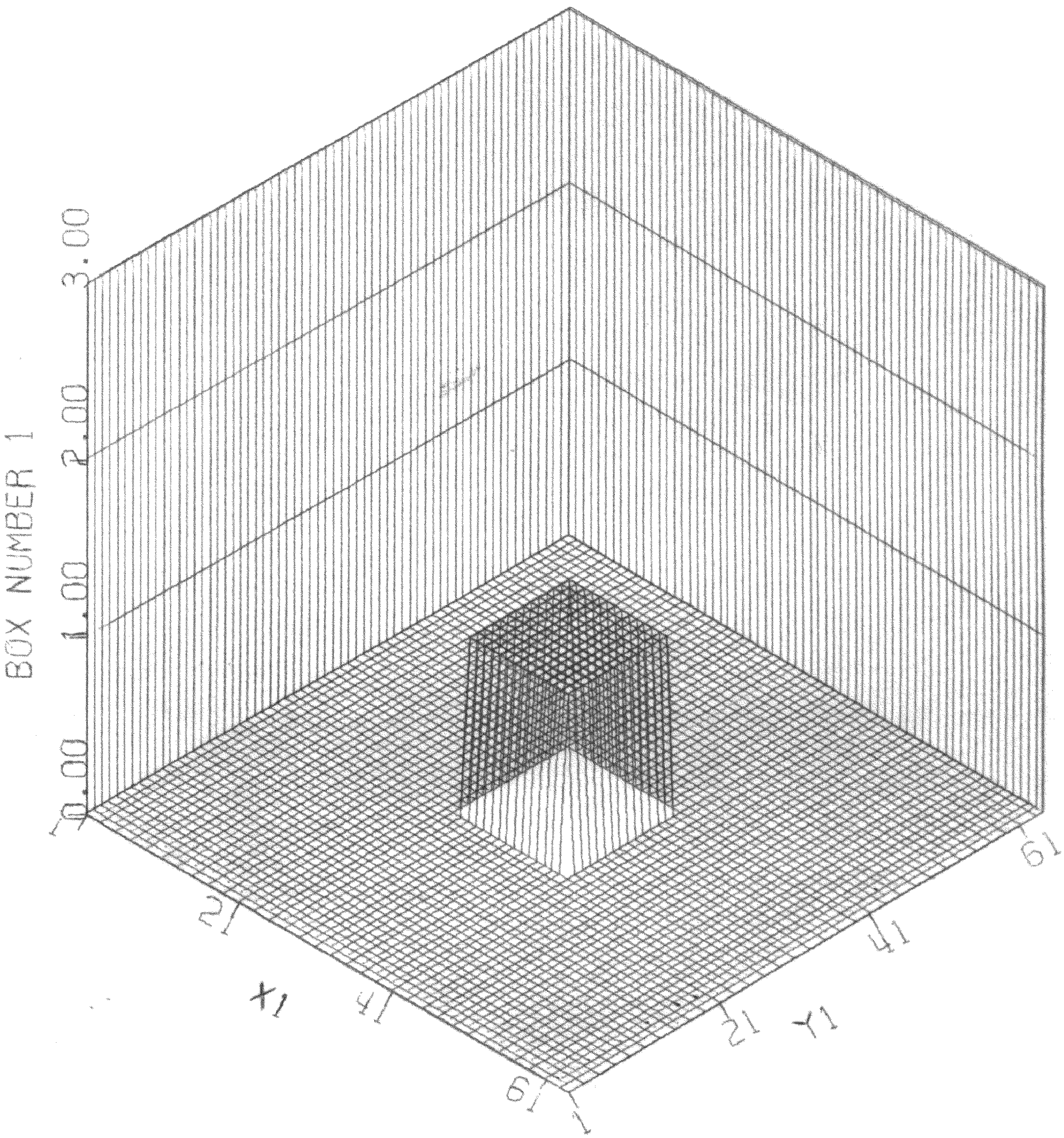


Figure 4-1 Box Number One Has Dimensions 14 by 14 by 1 and is Centered in a 64 by 64 Array of Zeros.

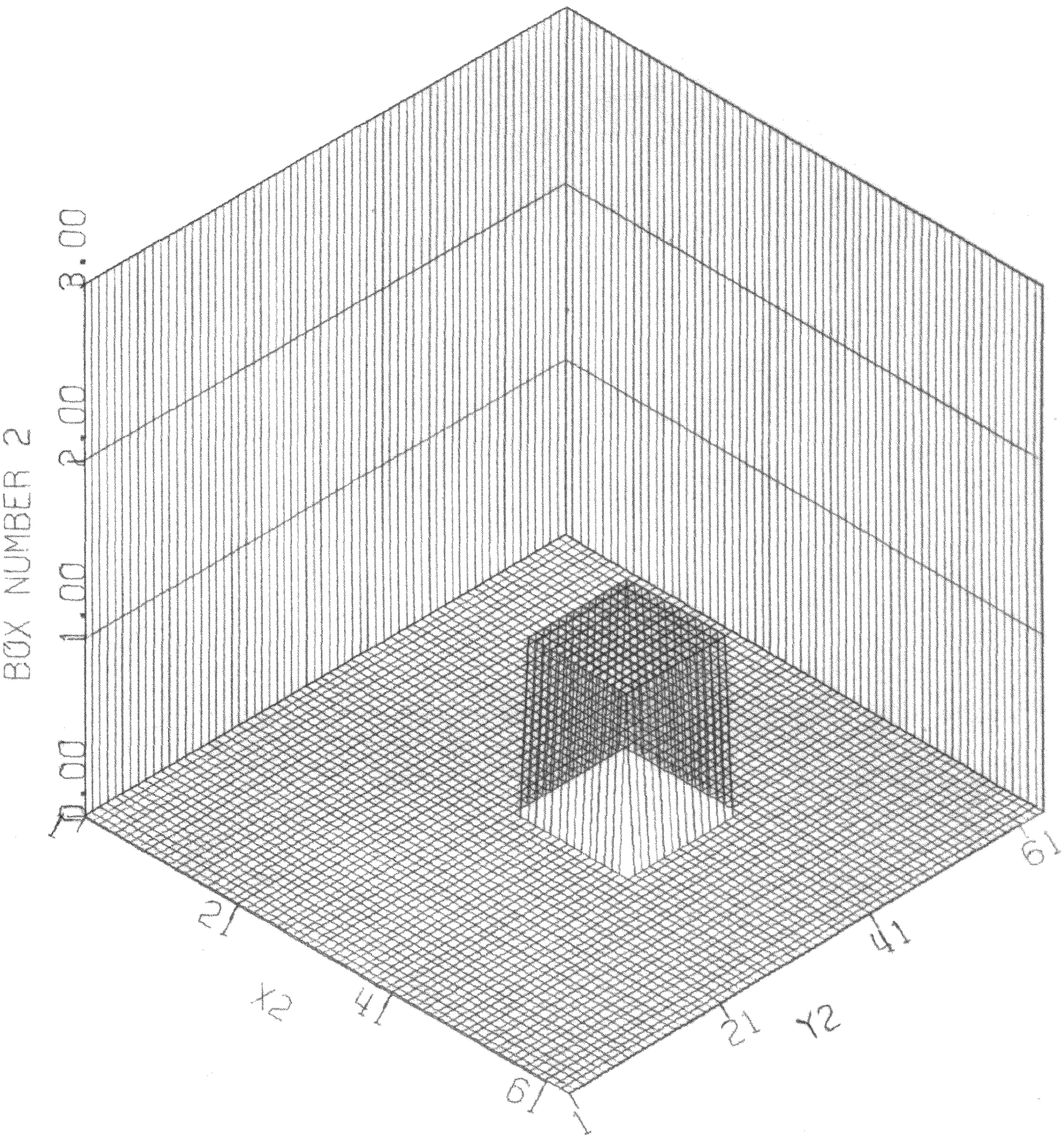


Figure 4-2 Box Number Two Has Dimensions 14 by 14 by 1 and is Offset From the Center of the 64 by 64 Array of Zeros by 4 in the X2 and 4 in the Y2 Directions.

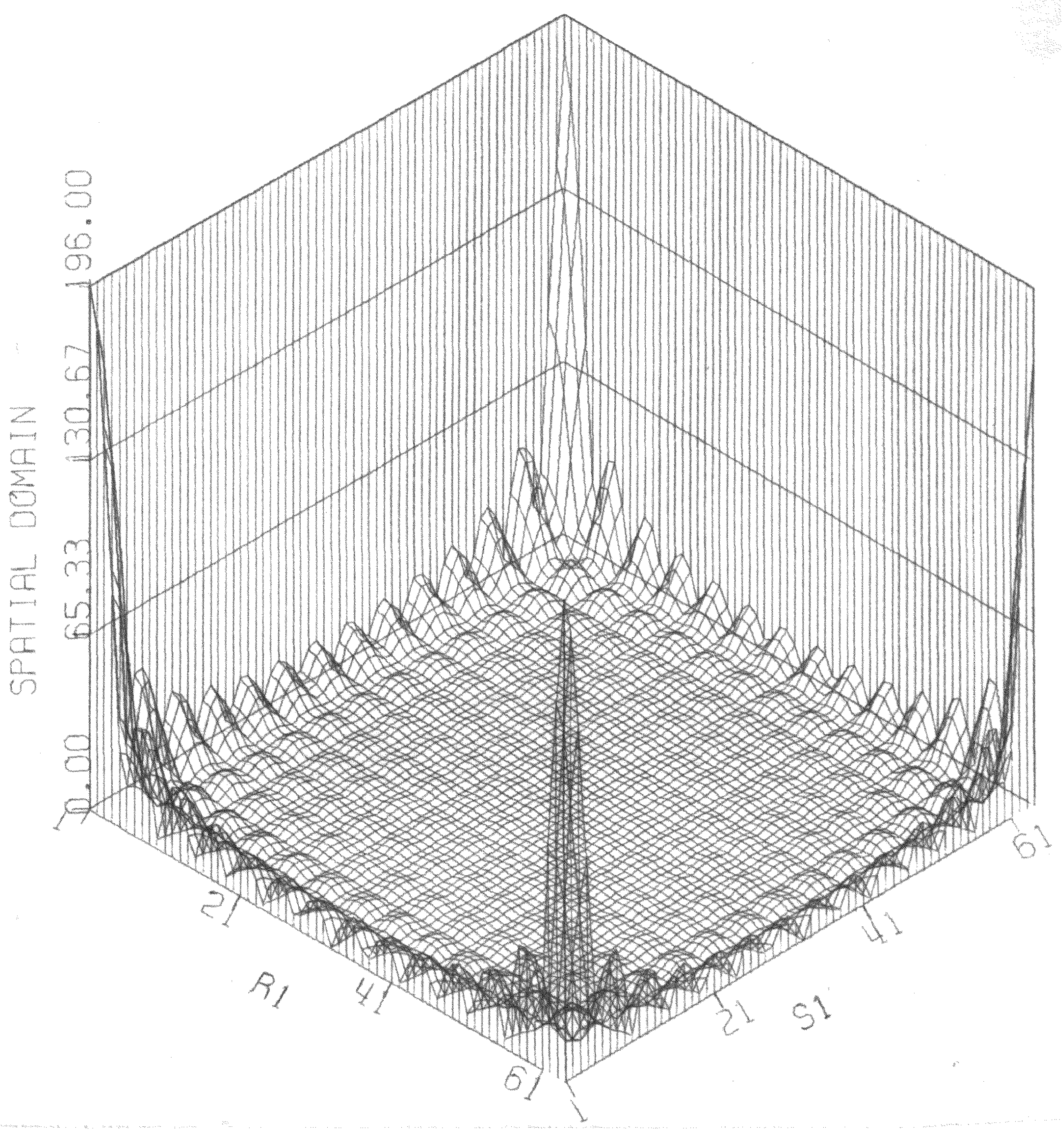


Figure 4-3 The Magnitude of the Two Dimensional Fourier Transform of Box Number One or Box Number Two.

array and so on. Interchanging quadrant 1 with quadrant 3 and quadrant 2 with quadrant 4 will result in the more familiar two dimensional sin (x)/x curve with the (0,0) spatial term located at (33, 33).

Since the boxes are located in an array of zeros, the actual image that is correlated can be of any size as long as the box is defined within it. For this example the correlating image (box number 2) will be of size 44 by 44 centered within the 64 by 64 array. With N = 44 and M = 64 the number of shifts will be M - N plus one for the zero shift or 21 shifts along each axis.

Before calculating the transforms, the mean of both images is removed. Then the variance of each is calculated and divided into the zero meaned images. This results in both images having a mean of zero and variance of one.

Using the fast Fourier transform, HARM,[17] the transform of both images is calculated. Each element of the transform of box number one is multiplied by the conjugate of the corresponding element in the transform of box number two. The inverse Fourier transform is then calculated of the resultant array. The final array, c, is an array of correlation coefficients of dimension 64 by 64. The (1, 1) coefficient represents the zero shift point, that is

$$c(1,1) = \Sigma\ \Sigma\ A(\alpha,\beta)\beta(0+\alpha,0+\beta).$$

The subscripts to the left of the equal sign are in FORTRAN array notation and subscripts to the right are part of the mathematical notation for defining correlation as previously defined. The c(1, 2) is the correlation coefficient with the second image offset by one shift in the Y direction. The c(1, 11) coefficient is offset by ten shifts in the Y direction. The c(1, 12) is a false correlation value because shifting the correlating image by 11 causes the image to begin repeating itself on the other side of the array. True correlation values are not resumed until the correlating image does completely reconstruct itself on the other side of the array at c(1, 55) or a -10 shifts in the Y direction. Continuing further, the c(1, 64) element is the correlation coefficient for a -1 shift in the Y direction.

Continuing this discussion in both the X and Y directions, it becomes apparent that the only valid correlation coefficients are in the corners of the array c(i, j).

quadrant I) $i, j = 1, \ldots, \frac{M-N}{2} + 1$

quadrant II) $i = \frac{M+N}{2} + 1, \ldots, M; \; j = 1, \ldots, \frac{M-N}{2} + 1$

quadrant III) $i,j = \frac{M+N}{2} + 1, \ldots, M$

quadrant IV) $i = 1, \ldots, \frac{M-N}{2} + 1; \; j = \frac{M+N}{2} + 1, \ldots, M.$

Removing the corners and positioning them in another array such that the c(1, 1) coefficient is in the center and the remaining coefficients are in the quadrants as indicated by Figure 4-4, results in an array of meaningful correlation coefficients as shown in Figure 4-5.

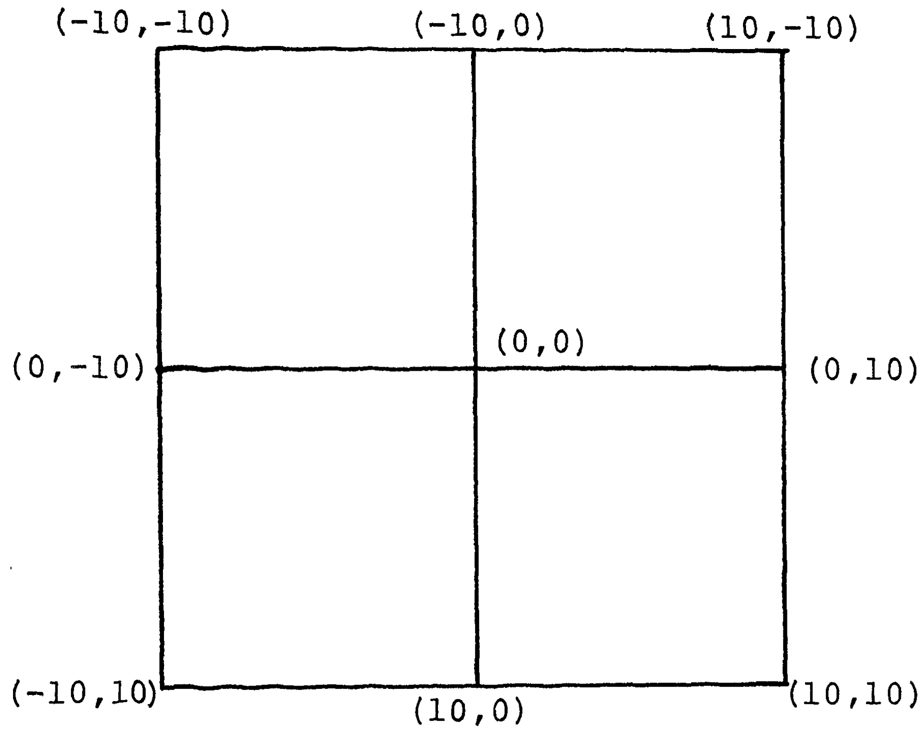


Figure 4-4 Repositioning Location for the Valid Correlation Coefficients.

From this figure the maximum correlation is at (-4, -4). Therefore, to align the two boxes, box number 2 must be moved four units in the -X direction and four units in the -Y direction.

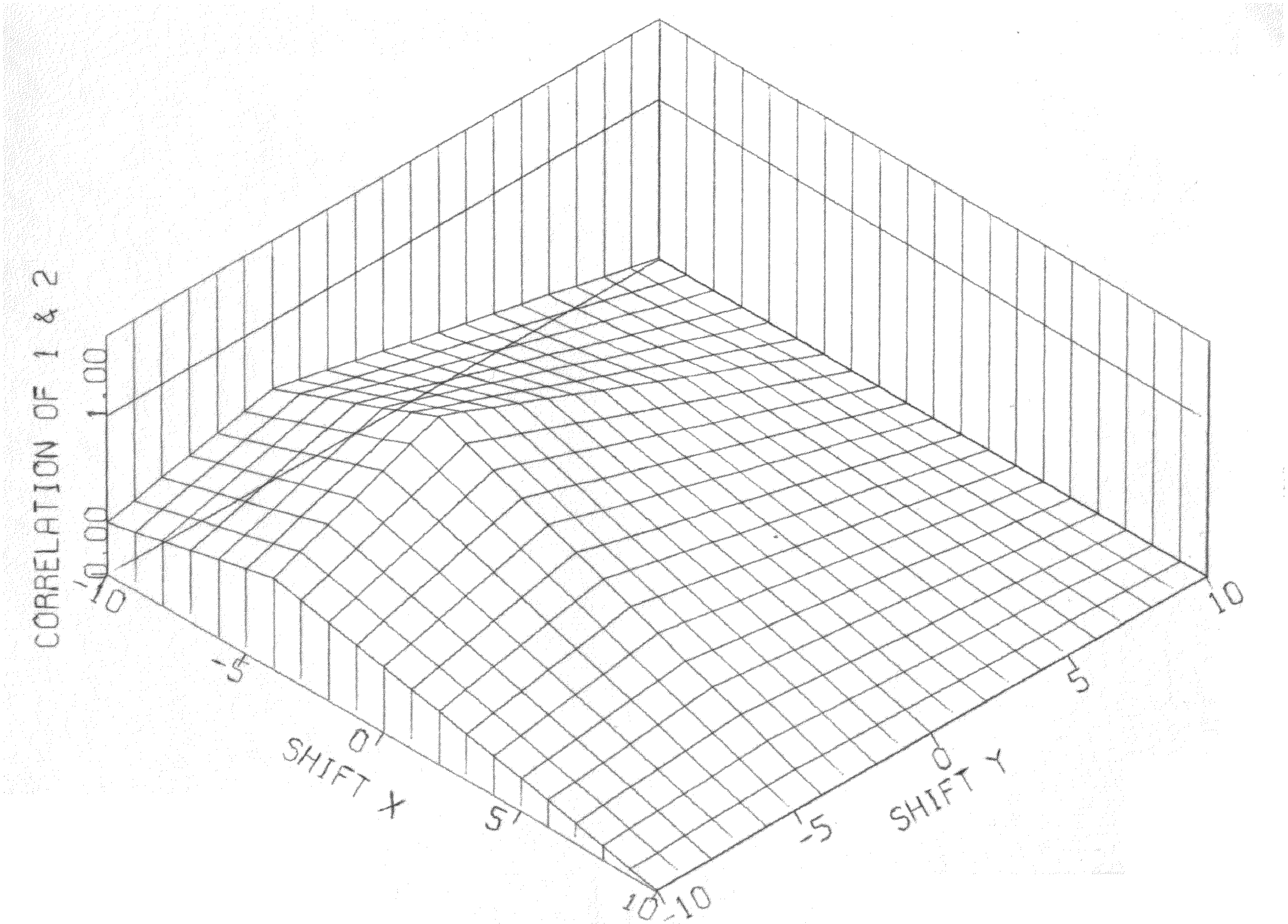


Figure 4-5 Array of Valid Correlation Coefficients Showing Location of Maximum Correlation at (-4,-4).

V. IMPLEMENTATION OF THE REGISTRATION SYSTEM

Because of core requirements, CPU time and operator instructions, the registration system implemented here consists of several programs which results in the complete registration of one picture with respect to another. The pictures are both stored on 9 track magnetic tapes as 900 scan lines. Each scan line represents 2000 picture elements of gray scale levels 1 to 64. The programs used in the registration system are outlined below in the order of their use.

a. The correlation program.

b. Least squares determination of the rotation angle.

c. Rotation program.

d. Modified "short version" correlation program.

e. Linear shift in X and Y directions.

A listing of the programs plus a flow chart description of their operation may be found in the Appendix.

A. The Correlation Program

In order to obtain information relative to the rotation and translation of the two images with respect to each other, more than one two dimensional correlation must be performed. Ideally, if the correlations were performed across the face of the picture, all the correlation coefficients would plot along a skewed straight line. To obtain the skewed line, this program calculates a maximum correlation coefficient for every twenty picture elements across the face of the picture. Since each row is 2000 elements long and the size of the base image is 64 elements, the program calculates the integer of (2000-64)/20 plus 1 or 97 coefficients per row.

The input to the correlation program is a data deck. Each card specifies the starting line where the multiple correlations are to be performed. There is no limit to the number of starting lines in the data deck; but since each row of multiple correlations requires about forty-five minutes of execution time, it is advised that the number of lines be kept to a minimum. In most cases one row is sufficient.

The correlation coefficients are calculated as described in detail in the previous chapter. The procedure is outlined briefly as follows:

1. Image A is 64 by 64 (M=64)
2. Image B is 36 by 36 (N=36)
3. Subtract mean from A and B
4. Normalize variance of A and B
5. Pad B with zeros on all sides to make it 64 by 64
6. Calculate FFT of A and B
7. $R(i,j)=A(i,j)B^*(i,j)$ for all $i,j=1,\ldots,64$; $*$=conjugate
8. $C=FFT^{-1}(R)$
9. Store corners of C in M-N+1 by M-N+1 array (29 by 29)
10. Locate maximum in M-N+1 by M-N+1 array

11. Relate location of maximum to present location of system and save.

This process is repeated for each of the 97 coefficients in the row.

The program outputs include a punched deck, a printer plot, and a listing. For each correlation a card is punched. Each card contains information as follows: the row and column the system is presently working on, the row and column location of the maximum correlation coefficient with respect to the system, and the value of the correlation coefficient. The printer plot portrays graphically the location of the 97 maximum correlation coefficients. Information obtained from this plot is used in the least squares program to determine the range over which the coefficients are linear. The listing contains the same information as that of the cards. This listing is useful as a visual aid in determining which coefficients are correct by their magnitude.

B. Least Squares Determination of the Rotation Angle

The rotation angle is found by a least squares analysis of the data punched from the correlation program. The program uses the values of (y_i,x_i) representing the row and column locations of the correlation maximums to find the n^{th} order polynomial relationship between x and y, that is

$$y = F(x) = a_o+a_1x+a_2x^2+ \ldots \qquad (5.1)$$

As input to the program the operator determines from the printer plot the approximate range over which the correlation coefficients are linear. This helps to minimize the number of bad correlation points in the least squares analysis, and gives a better estimate of F(x). Another restriction placed upon the input data is that only the locations with correlation values greater than 0.4 are considered. The order of the least squares equation may also be specified. It is recommended that the order of the polynomial be at least two. Then the a_2 term can be compared with the a_1 term to give an indication of any nonlinearity between the two images. If the ratio of a_1 to a_2 is small, then y becomes a linear function of x ($y = a_o+a_1x$), and a_1 is the slope of the correlation line.

Some indication of the number of rows needed for the rotation program is also found by

$$K = 2000\sin(\theta)+900\cos(\theta)-900 \qquad (5.2)$$

where $\theta = |\arctan(a_1)|$ is the angle of rotation of the correlation line. Two thousand is the length of each line, and 900 is the number of lines in the image. If K is greater than 35, then the dimension statements in the rotation program must be changed.

C. Rotation Program

This program rotates the picture by θ degrees, where

$$\theta = |\arctan(a_1)|. \tag{5.3}$$

The rotation is accomplished by the linear transformation

$$\begin{aligned} x' &= x\cos(\theta) - y\sin(\theta) \\ y' &= x\sin(\theta) + y\cos(\theta) \\ x &= 1,\ldots,2000 \text{ and } y = 1,\ldots,900 \end{aligned} \tag{5.4}$$

where x and y are element locations within the rotated image and x' and y' are the corresponding points within the unrotated image. Since the unrotated image is also 900 by 2000, there will be some values of x' and y' which will not correspond to any point within the unrotated image. When this occurs, a zero is stored in the location of x and y. As input, the program reads the absolute value of a_1. If a_1 is negative, then the correlating image, B, must be rotated. For a 900 line image, this program requires approximately seventy-five minutes of computer time.

D. Modified "Short Version" Correlation Program

This program uses the rotated image to calculate the two dimensional correlation coefficients in the same manner as the first correlation program, but in increments of 100 picture elements instead of 20. The program requires about nine minutes of execution time per row. Output of the program is a data deck as described in the correlation program plus the listing of the row and column shifts for each correlation. The average row and column shift can be determined by scanning the row and column shifts.

E. Linear Shift in X and Y Directions

The linear shift program pads the rows and columns of the rotated picture with zeros. The input consists of the average row and column shifts determined by the modified correlation program.

VI. RESULTS

In this chapter the results of registering a digitized color image (the base image) with a color infrared image (the correlating image), Figures 3-1 and 3-2, using the system of program described in chapter five, is discussed. The output of each program is discussed, along with its relationship to the next program.

Figure 6-1 is a plot of the location of the correlation maximums in the correlation program. The range of the good correlation values is obvious. With a few exceptions, the good values are located in a straight line between 0 and 1287. The exceptions, as is shown in the listing in the Appendix, mostly have low correlation values.

In the least squares program the straight line segment over the range indicated above is specified as part of the input data. The order of the polynomial fit is two. The remainder of the data is supplied by the punched output of the correlation program. The least squares program second order polynomial fit is

$$y = F(x) = a_o+a_1x+a_2x^2 \qquad (6.1)$$

and was found to be

$$y = F(x) = 3.711502+(-0.01439846)X+(0.1192093E-06)X^2.$$

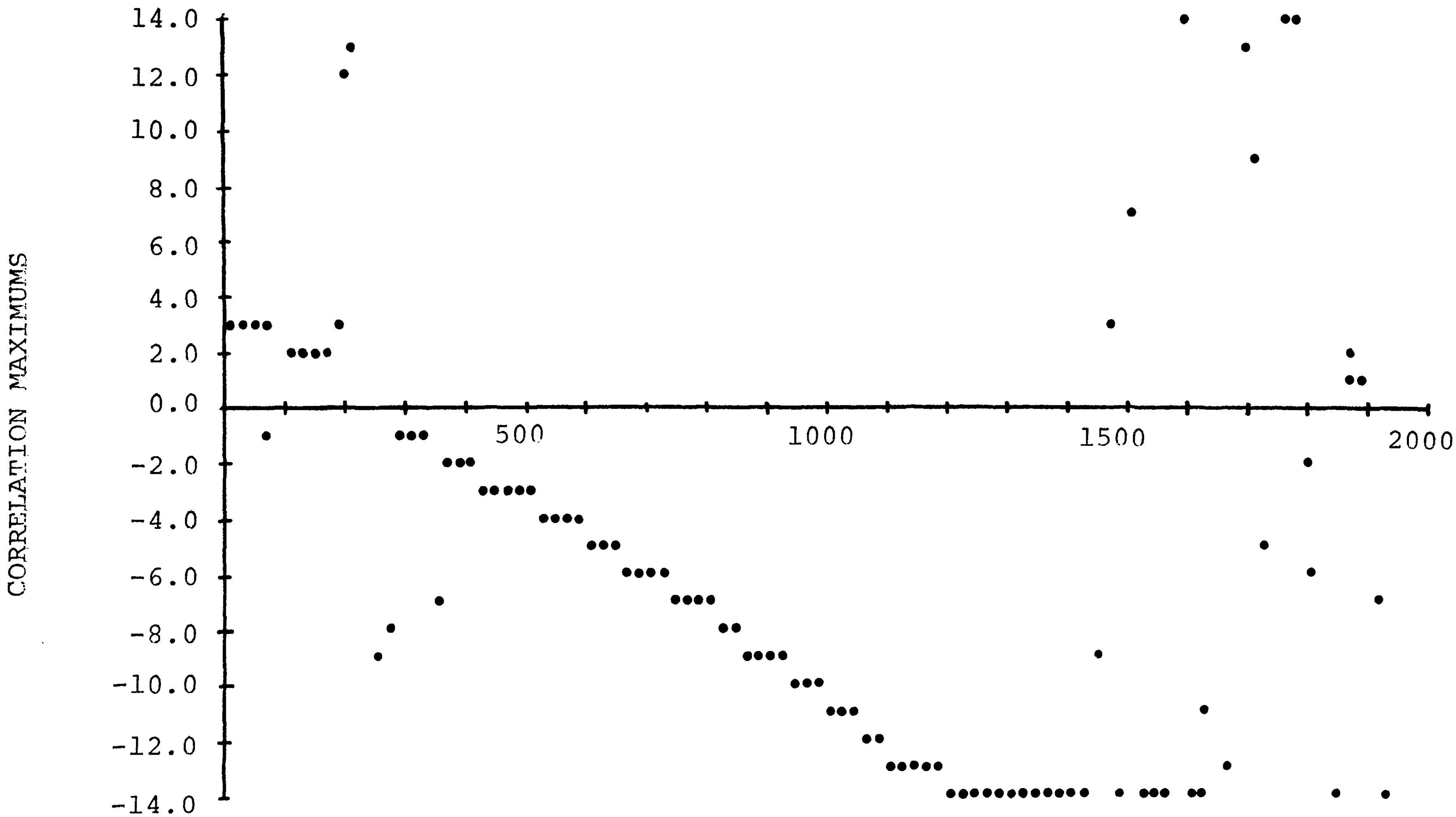


Figure 6-1 The Points of Maximum Correlation Between the Color and Color Infrared Images.

The ratio of a_2 to a_1 is of magnitude 10^{-3}. Therefore, the two images are linearly translated and rotated. Also in the output of this program is an indication of the number of rows which must be built up in the rotation program in order for the rotation to take place. If this number is greater than 35, the dimension statement in the rotation program must be changed to the value indicated. For the color and color infrared images the number of rows is 34.

Since a_1 is negative, the correlating image, color infrared, must be rotated. The coefficient a_1 is the input to the rotation program. The output is a new magnetic tape containing the rotated color infrared image.

After the two images are rotated, the modified correlation program is run to check the results of the rotation and find the linear translational component of the images. This program performs the correlation calculations in the same manner as the first correlation program, but in intervals of one hundred picture elements instead of twenty. Depending on how good the rotation was, the printer plot may or may not be produced. (To produce a printer plot the maximum and minimum of the dependent variable, row shifts, cannot be the same. If this occurs, as indeed we hope it will, no plot is produced.) However, by scanning the output row and column shifts along with their respective correlation coefficients, an average row and column shift can be found. The author feels that this average is better than an analytical average because the reader will weigh the row and column locations by their respective correlation coefficients in deciding which ones represent the average. For example, three or four equal row and column shifts with correspondingly high correlation coefficients (greater than 0.7) would have more weight in the row and column shift average than the same number of shifts with correlation coefficients in the 0.4 to 0.6 range. From the listing as shown in the Appendix, the row and column shift was determined to be 4 and 8 respectively. Therefore, the correlating image must be moved forward 4 rows and to the right 8 columns. This is accomplished in the linear shift program, and results in another new tape containing the rotated and translated image.

VII. CONCLUSION

This investigation has touched briefly on the topic of multispectral classification of images. Before any meaningful classification can be accomplished, the physical location of a point within an image must be addressable in all images. Digitally registered images are of primary concern in a multispectral classification system.

Registration of two images requires knowledge of the rotational and translational components by which the two images are misaligned. It has been shown that using the fast Fourier transform to find the correlation between the two images is a method of calculating the misalignment vectors. By using the five programs developed in this investigation, a step by step procedure leads to the alignment of the images. The programs developed can be implemented on any computer with medium core storage (220 k bytes), two tape drives, and disk capabilities. The time requirements to perform the registration are somewhat excessive. A minimum of approximately 140 minutes of execution time is required to perform the registration. The largest part of this is the 75 minutes needed for the rotation program.

BIBLIOGRAPHY

1. B. G. Wixson and W. H. Tranter, "An Investigation of Environmental Pollution by Lead and Other Heavy Metals from Industrial Development in Southeastern Missouri," Proceedings of the 1972 IEEE National Telecommunications Conference, Dec., 1972.
2. G. W. North, "Remote Sensing of Environmental Pollution," Proceedings of the 1971 International Workshop on Earth Resources Survey Systems, Vol. II, May 3-14, 1971, Washington: Government Printing Office, 1971, pp. 291-301.
3. L. R. Pettinger, "Field Data Collection - An Essential Element in Remote Sensing Applications," Proceedings of the 1971 International Workshop on Earth Resources Survey Systems, Vol. II, May 3-14, 1971, Washington: Government Printing Office, 1971, pp. 49-64.
4. M. R. Holter and W. L. Wolfe, "Optical-Mechanical Scanning Techniques," Proceedings IRE, Vol. 47, Sept., 1959, pp. 1546-1550.
5. R. N. Colwell, "The Future for Remote Sensing of Agricultural, Forest and Range Resources," Committee on Science and Astronautics, U. S. House of Representatives. Remote Sensing of Earth Resources, Washington: Government Printing Office, 1972, pp. 103-117.

6. P. E. Anuta, "Spatial Registration of Multispectral and Multitemporal Digital Imagery Using Fast Fourier Transform Techniques," IEEE Transactions on Geoscience Electronics, Vol. IV, Oct., 1970, pp. 353-368.

7. J. L. Clapp, R. W. Kiefer, M. M. McCarthy and B. J. Niemann, Jr., "Interdisciplinary Research on the Evaluation of ERTS - A Data to the Regional Land Use Planning Process," Eighth International Symposium on Remote Sensing of Environment, Oct. 26-, 1972, Center for Remote Sensing Information and Analysis, The University of Michigan, 1972, pp. 144-145.

8. Richard S. Williams, Jr., "Satellite Geological and Geophysical Remote Sensing of Iceland," Eighth International Symposium on Remote Sensing of Environment, Oct. 2-6, 1972, Center for Remote Sensing Information and Analysis, The University of Michigan, 1972, pp. 146-147.

9. William C. Draeger and Andrew S. Benson, "Applications of ERTS - A Imagery to Agricultural Resource Evaluation," Eighth International Symposium on Remote Sensing of Environment, Oct. 2-6, 1972, Center for Remote Sensing Information and Analysis, The University of Michigan, 1972, pp. 148-149.

10. George E. Stoertz and George E. Erickson, "Expected Results of Hydrologic and Geologic Studies Using ERTS Imagery of the Atacama Desert, Altiplano, and Puna De Atacama (Southwest Bolivia, Northwest Argentina, and Northern Chile)," Eighth International Symposium on

Remote Sensing of Environment, Oct. 2-6, 1972, Center for Remote Sensing Information and Analysis, The University of Michigan, 1972, pp. 156-157.

11. "Remote Multispectral Sensing in Agriculture," Laboratory for Agricultural Remote Sensing, Annual Report, 1965-1969, 4 Vols., Purdue University, Agricultural Experiment Station, Lafayette, Indiana.

12. P. E. Anuta, T. L. Phillips and D. A. Landgrebe, "Data Handling and Analysis of the 1971 Corn Blight Watch Experiment," Proceedings of the 1972 IEEE National Telecommunications Conference, Dec., 1972.

13. J. W. Hilty and B. S. Ausmus, "Multispectral Photography for the Detection of Diseased Corn," Proceedings of the 10th Annual IEEE Region 3 Convention, April, 1971.

14. D. J. Popp, D. S. McCormack and G. M. Lee, "Digital Image Registration by Correlation Techniques," Proceedings of the 1972 IEEE National Telecommunications Conference, Dec., 1972.

15. J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Math. Comput., Vol. XIX, April, 1965, pp. 297-301.

16. R. S. Simpson and R. G. Houts, Fundamentals of Analog and Digital Communication Systems (Boston: Allyn and Bacon, Inc., 1971), pp. 72-73.

17. "System/360 Scientific Subroutine Package (360A-CM-03X) Version III Programmer's Manual," IBM Application Program H20-0205-3, 1968, pp. 276-279.

VITA

Victor F. Wilreker, Jr., was born on November 5, 1943, in Washington, D.C. He received his primary and secondary education in Baltimore, Maryland. He has received his college education from the University of Maryland, College Park, Maryland; Loyola Evening College, Baltimore, Maryland; Johns Hopkins Evening College, Baltimore, Maryland; St. Mary's Evening College, St. Mary's City, Maryland; Catonsville Community College, Catonsville, Maryland; and the University of Missouri - Rolla, Rolla, Missouri. He received a Bachelor of Science degree in Electrical Engineering from the University of Missouri - Rolla, Rolla, Missouri in July of 1971.

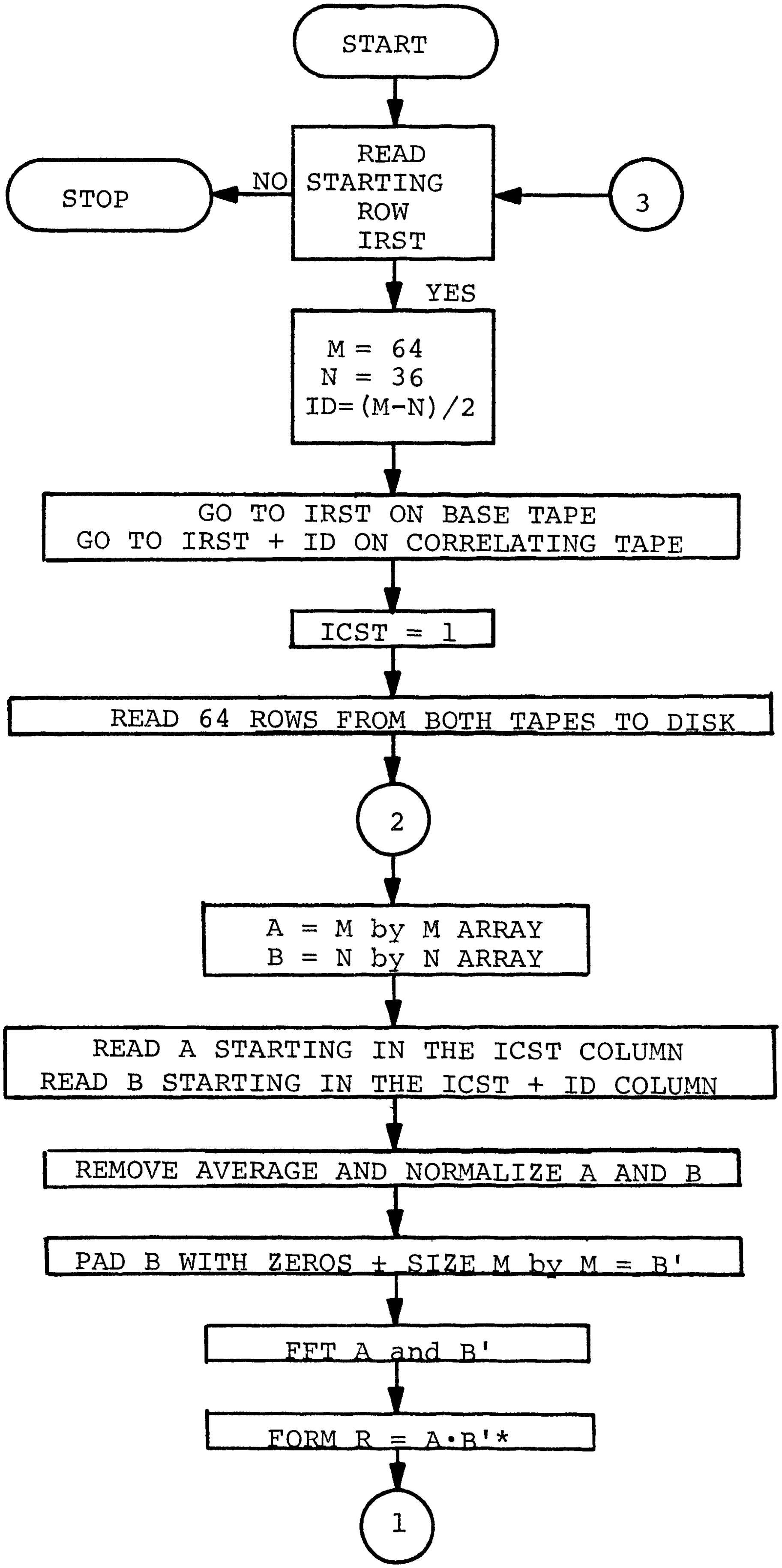
Mr. Wilreker has been enrolled in the Graduate School of the University of Missouri - Rolla since August of 1971 and has been a Graduate Teaching Assistant in the Electrical Engineering Department at the University of Missouri - Rolla for the period of August of 1971 to the present.

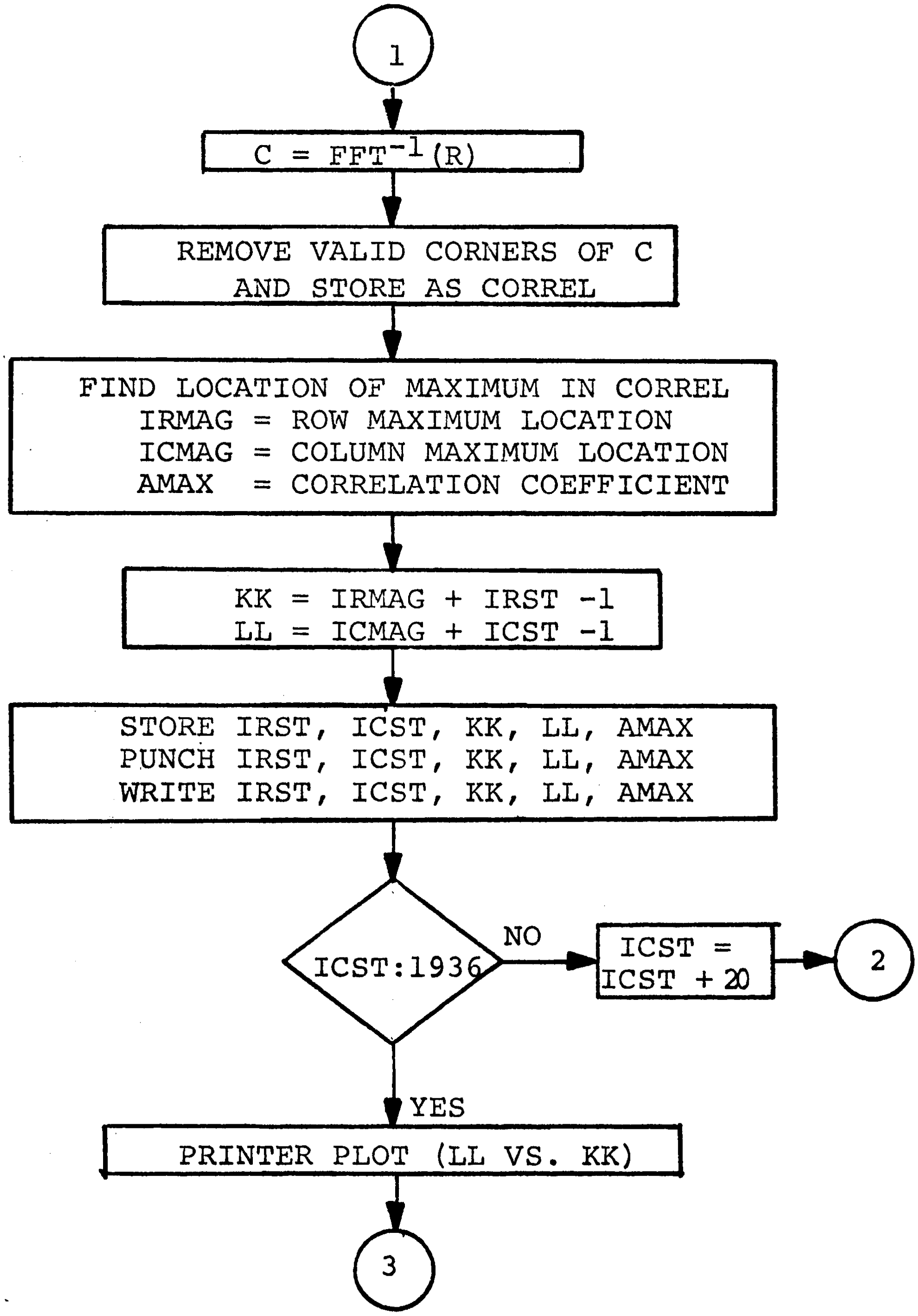
The author is a member of Eta Kappa Nu and Tau Beta Pi, an associate member of the Society of Sigma Xi, and a student member of the IEEE Communications Society and the IEEE Power Engineering Society. He was awarded the Chemical Rubber Company Freshman Chemistry Award at Catonsville Community College, Catonsville, Maryland in September of 1969.

APPENDIX A

FLOW CHART

THE CORRELATION PROGRAM


START
READ STARTING ROW IRST
NO
STOP
3
YES
M = 64
N = 36
ID=(M-N)/2
GO TO IRST ON BASE TAPE
GO TO IRST + ID ON CORRELATING TAPE
ICST = 1
READ 64 ROWS FROM BOTH TAPES TO DISK
2
A = M by M ARRAY
B = N by N ARRAY
READ A STARTING IN THE ICST COLUMN
READ B STARTING IN THE ICST + ID COLUMN
REMOVE AVERAGE AND NORMALIZE A AND B
PAD B WITH ZEROS + SIZE M by M = B'
FFT A and B'
FORM R = A•B'*
1



1
C = FFT^-1(R)
REMOVE VALID CORNERS OF C AND STORE AS CORREL
FIND LOCATION OF MAXIMUM IN CORREL
IRMAG = ROW MAXIMUM LOCATION
ICMAG = COLUMN MAXIMUM LOCATION
AMAX = CORRELATION COEFFICIENT
KK = IRMAG + IRST -1
LL = ICMAG + ICST -1
STORE IRST, ICST, KK, LL, AMAX
PUNCH IRST, ICST, KK, LL, AMAX
WRITE IRST, ICST, KK, LL, AMAX
ICST:1936
NO
ICST = ICST + 20
2
YES
PRINTER PLOT (LL VS. KK)
3

```
//OS    JOB CARD
//* LIMITS=(T=120,P=100,R=(230),C=50)
//S1 EXEC RFTGCLG
//C.SOURCE DD *
C
C ******* THE CORRELATION PROGRAM ***********
C
      INTEGER COLOR/6/,INFRA/5/,COLR/8/,IFR/9/,N1(3)/6,6,0/,INVER(16)
      INTEGER KK(100),LL(100)
      INTEGER*2 INUM(2000)
      REAL XKK(100),YLL(100)
      REAL DUMMY(64,64),S(16),CORREL(32,32),RE(100)
      COMPLEX X(64,64,1),Y(64,64,1),R(64,64,1)
      COMMON/WORK/X
      COMMON/WORKC/DUMMY
      COMMON/WORKD/R
      COMMON/WORKE/Y
      COMMON/WORKF/CORREL
  201 FORMAT('0',6E15.3/(' ',6E15.3))
  202 FORMAT('0',6F15.5/(' ',6F15.5))
      M=64
      N=36
      ID=(M-N)/2
  230 FORMAT('1')
    5 READ(1,200,END=999)IRST
  200 FORMAT(I3)
      IW=IRST-1
      INCR=0
      WRITE(3,230)
      IF(IRST-1) 8,8,7
    7 IR=IRST-1
      DO 6 I=1,IR
      READ(COLOR) INUM
    6 READ(INFRA) INUM
    8 DO 4 I=1,ID
```

```
  4 READ(INFRA) INUM
    DO 10 I=1,64
    READ(COLOR) INUM
    WRITE(COLR) INUM
    READ(INFRA) INUM
 10 WRITE(IFR) INUM
    REWIND COLR
    REWIND IFR
    REWIND COLOR
    REWIND INFRA
    DO 99 ICST=1,1935,20
    DO 11 I=1,64
    DO 11 J=1,64
 11 DUMMY(I,J)=0.0
    CALL LOAD(M,COLR,ICST)
    A=AVG(M)
    CALL SSUB(DUMMY,A,DUMMY,M,M,0)
    B=ROOTSQ(M)
    CALL SDIV(DUMMY,B,DUMMY,M,M,0)
    CALL DTOX(M)
    DO 12 I=1,64
    DO 12 J=1,64
 12 DUMMY(I,J)=0.0
    ICS=ICST+ID
    CALL LOAD(N,IFR,ICS)
    A=AVG(N)
    CALL SSUB(DUMMY,A,DUMMY,M,M,0)
    B=ROOTSQ(N)
    CALL SDIV(DUMMY,B,DUMMY,M,M,0)
300 FORMAT(8G10.3)
    CALL DTOY(M,N)
    CALL HARM(X,N1,INVER,S,1,IFER1)
    CALL HARM(Y,N1,INVER,S,1,IFER2)
    CALL MULTI(M)
 13 CONTINUE
```

```
      CALL HARM(R,N1,INVER,S,-1,IFER3)
      CALL MAGI(M)
      CALL CORATE(M,N,ID)
      CALL MAXMIN(ID,IRMAG,ICMAG,ADSMIN,AMAX)
      INCR=INCR + 1
      CMIN=0.0
      IF=4
      RE(INCR)=AMAX
      KK(INCR)=IRMAG + IRST - 1
      LL(INCR)=ICMAG + ICST - 1
      IB=ICST-1
      WRITE(2,310) IW,IB,KK(INCR),LL(INCR),AMAX
  310 FORMAT(4I10,G15.7)
   99 CONTINUE
   30 DO 31 I=1,INCR
   31 WRITE(3,215) KK(I),LL(I),RE(I)
  215 FORMAT('0',2I10,E15.5)
      DO 21 I=1,INCR
      XKK(I)=FLOAT(KK(I))
   21 YLL(I)=FLOAT(LL(I))
      CALL PPLT(YLL,XKK,INCR)
      DO 899 I=1,20
  899 WRITE(2,310)
      GO TO 5
  999 STOP
      END

      SUBROUTINE MULTI(M)
C
C
C ******** THIS PROGRAM COMPUTES A RESULTANT ARRAY BY MULTIPLYING
C          THE BASE ARRAY BY THE COMPLEX CONJUGATE OF THE CORRELATING ARRAY
C        THE INPUTS ARE*
C                 M=THE SIZE OF THE BASE MATRIX
C        COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
```

```
C
C
      COMPLEX*8 X(64,64,1),Y(64,64,1),R(64,64,1)
      COMPLEX*8 CONJG
      COMMON/WORKE/Y
      COMMON/WORKD/R
      COMMON/WORK/X
      DO 10 I=1,M
      DO 10 J=1,M
   10 R(I,J,1)=X(I,J,1)*CONJG(Y(I,J,1))
      RETURN
      END

      SUBROUTINE MAGI(M)
C
C
C********THIS SUBROUTINE COMPUTES THE MAGNITUDE OF THE COMPLEX M-BY-M ARRAY
C        THE INPUTS ARE*
C                 M=THE SIZE OF THE BASE MATRIX
C        COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
C
C
      COMPLEX*8 R(64,64,1)
      REAL*4 PI(64,64)
      COMMON/WORKD/R
      COMMON/WORKC/PI
      DO 10 I=1,M
      DO 10 J=1,M
   10 PI(I,J)=REAL(R(I,J,1))
      RETURN
      END

      SUBROUTINE CORATE(M,N,DEL)
C
C
```

```
C********THIS SUBROUTINE OBTAINS THE  NONCYCLIC CORRELATION MATRIX FROM
C        CYCLIC M-BY-M INVERSE TRANSFORM.  THE INPUTS ARE*
C                 M=SIZE OF THE BASE MATRIX
C        N=THE SIZE OF THE CORRELATING MATRIX
C         DEL=THE SIZE OF THE NONCYCLIC CORRELATING MATRIX.  ITS VALUE
C                  IS 2*(NUMBER OF SHIFTS) + 1.
C        COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
C
C
      INTEGER DELTA,DEL
      REAL*4 PI(64,64),COR(32,32)
      COMMON/WORKC/PI
      COMMON/WORKF/COR
C             TEST COR
      DO 5 I=1,DEL
      DO 5 J=1,DEL
    5 COR(I,J)=0.0
      DELTA=((M-N)/2)-1
      IDELTA=DELTA+2
      DO 10 K=1,IDELTA
      DO 10 L=1,IDELTA
      IR=DELTA+K+1
      IC=DELTA+L+1
   10 COR(IR,IC)=PI(K,L)
      ID=M-IDELTA+2
      DO 20 K=1,IDELTA
      DO 20 L=ID,M
      IR=K+IDELTA-1
      IC=L-ID+1
   20 COR(IR,IC)=PI(K,L)
      ID=M-DELTA
      DO 30 K=ID,M
      DO 30 L=1,IDELTA
      IR=K-ID+1
      IC=L+DELTA+1
```

```
   30 COR(IR,IC)=PI(K,L)
      ID=M-DELTA
      DO 40 K=ID,M
      DO 40 L=ID,M
      IR=K-ID+1
      IC=L-ID+1
   40 COR(IR,IC)=PI(K,L)
      RETURN
      END

      SUBROUTINE MAXMIN(ID,IRMAG,ICMAG,ADSMAX,AMAX)
C ******** THIS SUBROUTINE FINDS THE LOCATION OF THE MAXIMUM WITHIN AN
C             ARRAY OF CORRELATION COEFFICIENTS.
C              INPUTS ARE*
C            ID=THE SIZE OF THE ARRAY OF CORRELATION COEFFICIENTS
C             OUTPUTS ARE*
C                IRMAG= THE ROW LOCATION OF THE MAXIMUM
C                ICMAG=THE COLUMN LOCATION OF THE MAXIMUM
C                ADSMAX= THE EUCLIDEAN DISTANCE TO THE MAXIMUM
C           AMAX=THE MAXIMUM CORRELATION COEFFICIENT
C          COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
      REAL COR(32,32)
      COMMON/WORKF/COR
      IDT=(ID*2)+1
      IRMAG=2
      ICMAG=2
      DO 10 I=1,IDT
      DO 10 J=1,IDT
      IF(COR(I,J).GT.COR(IRMAG,ICMAG)) GO TO 20
   10 CONTINUE
      GO TO 30
   20 IRMAG=I
      ICMAG=J
      GO TO 10
   30 AMAX=COR(IRMAG,ICMAG)
```

```
      IRMAG=IRMAG-ID-1
      ICMAG=ICMAG-ID-1
      A=FLOAT(IRMAG)
      B=FLOAT(ICMAG)
      ADMAX=SQRT(ABS(A)**2 + ABS(B)**2)
      RETURN
      DEBUG SUBCHK
      END

      SUBROUTINE LOAD(M,NAME,ICST)
C *******  THIS SUBROUTINE LOADS AN M BY M ARRAY FROM THE UNIT(NAME)
C          INPUTS ARE*
C            M= THE SIZE OF THE ARRAY TO BE LOADED INTO ACTIVE CORE
C            NAME= THE UNIT (DISK) FROM WHICH THE INFORMATION IS TO BE READ
C            ICST= THE STARTING COLUMN THAT IS TO BE READ
C        COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
      REAL DUMMY(64,64)
      INTEGER*2 INUM(2000)
      COMMON/WORKC/DUMMY
      DO 10 I=1,M
      N=ICST
      READ(NAME) INUM
      DO 20 J=1,M
      MET=INUM(N)
      DUMMY(I,J)=FLOAT(MET)
   20 N=N + 1
   10 CONTINUE
      REWIND NAME
      RETURN
      DEBUG SUBCHK
      END

      SUBROUTINE DTOY(M,N)
C
C
```

```
C********THIS SUBROUTINE CHANGES THE N-X-N INPUT ARRAY TO THE COMPLEX
C        M-X-M ARRAY WITH THE REMAINDER OF THE NON N-XN ARRAY FILLED
C         OUT WITH ZEROS.  THE INPUTS ARE*
C                 M=THE SIZE OF THE BASE INPUT ARRAY
C                 N=THE SIZE OF THE CORRELATING ARRAY.
C        COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
C
C
      COMPLEX*8 Y(64,64,1),CMPLX
      REAL*4 DUMMY(64,64)
      COMMON/WORKE/Y
      COMMON/WORKC/DUMMY
      DO 10 I=1,M
      DO 10 J=1,M
   10 Y(I,J,1)=(0.0,0.0)
      IMN=((M-N)/2)
      DO 20 I=1,N
      DO 20 J=1,N
      IC=IMN+J
      IR=IMN+I
   20 Y(IR,IC,1)=CMPLX(DUMMY(I,J),0.0)
      RETURN
      END

      SUBROUTINE DTOX(M)
C
C
C********SUBROUTINE CONVERTS THE BASE INPUT ARRAY INTO COMPLEX FORM.
C        THE INPUTS ARE*
C                 M=SIZE OF THE BASE INPUT ARRAY.
C        COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
C
C
      COMPLEX*8 X(64,64,1),CMPLX
      REAL*4 DUMMY(64,64)
```

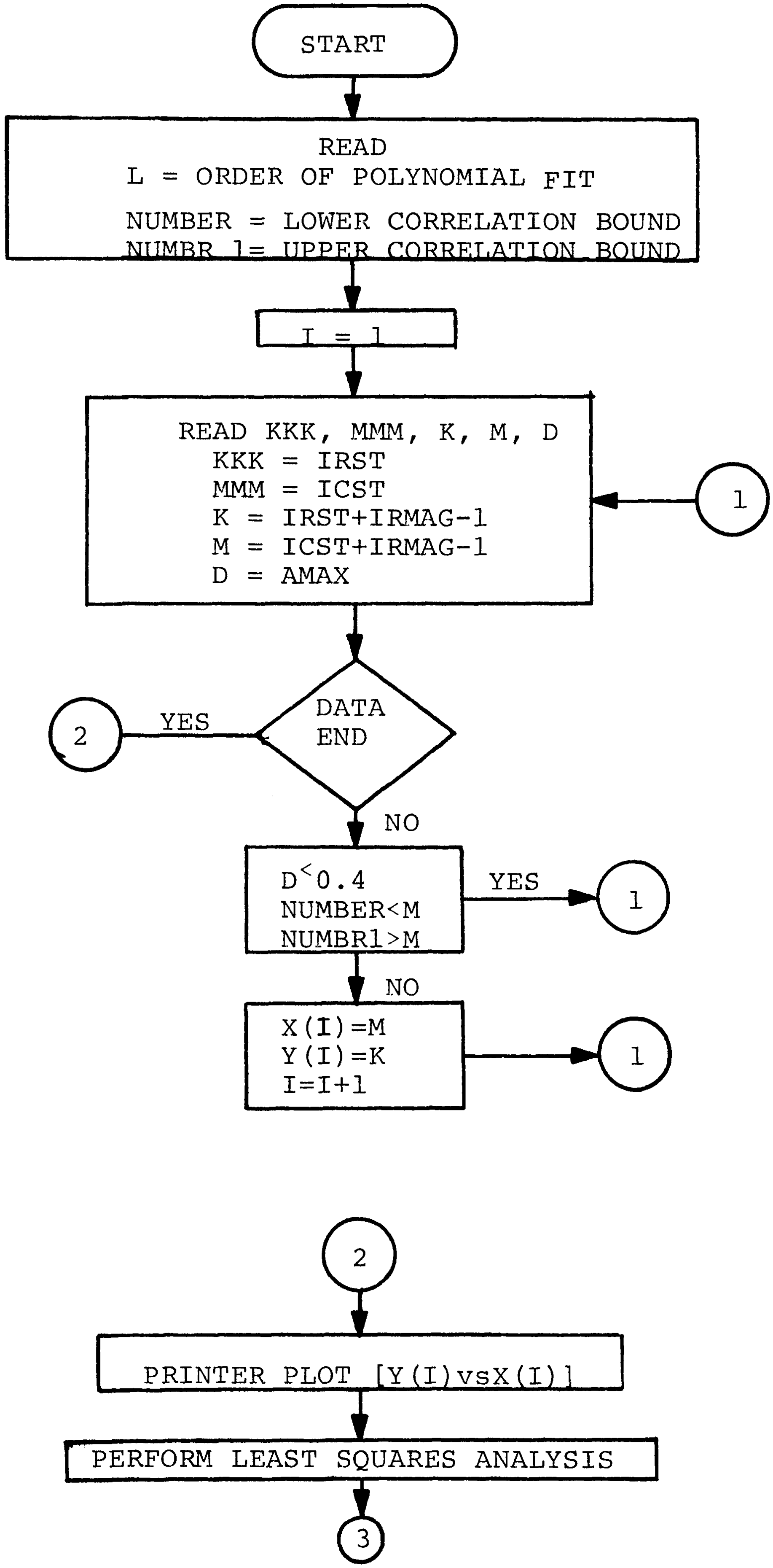
```
      COMMON/WORK/X
      COMMON/WORKC/DUMMY
      DO 10 I=1,M
      DO 10 J=1,M
   10 X(I,J,1)=CMPLX(DUMMY(I,J),0.0)
      RETURN
      END

      FUNCTION AVG(M)
C
C
C********* THIS FUNCTION SUBROUTINE CALCULATES THE STATISTICAL MEAN OF AN
C           M BY M ARRAY.
C           THE INPUTS ARE*
C                  M=SIZE OF THE INPUT ARRAY
C         COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
C
C
      REAL*4 DUMMY(64,64)
      COMMON/WORKC/DUMMY
      A=0.0
      DO 10 I=1,M
      DO 10 J=1,M
   10 A=A+DUMMY(I,J)
      AVG=A/(M*M)
      RETURN
      END

      FUNCTION ROOTSQ(M)
C
C
C********THIS FUNCTION SUBROUTINE CALCULATES THE ROOT SUM OF THE SQUARES
C         OF THE SAMPLE SPACE.  INPUTS ARE*
C                  M=SIZE OF THE INPUT ARRAY
C         COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
```

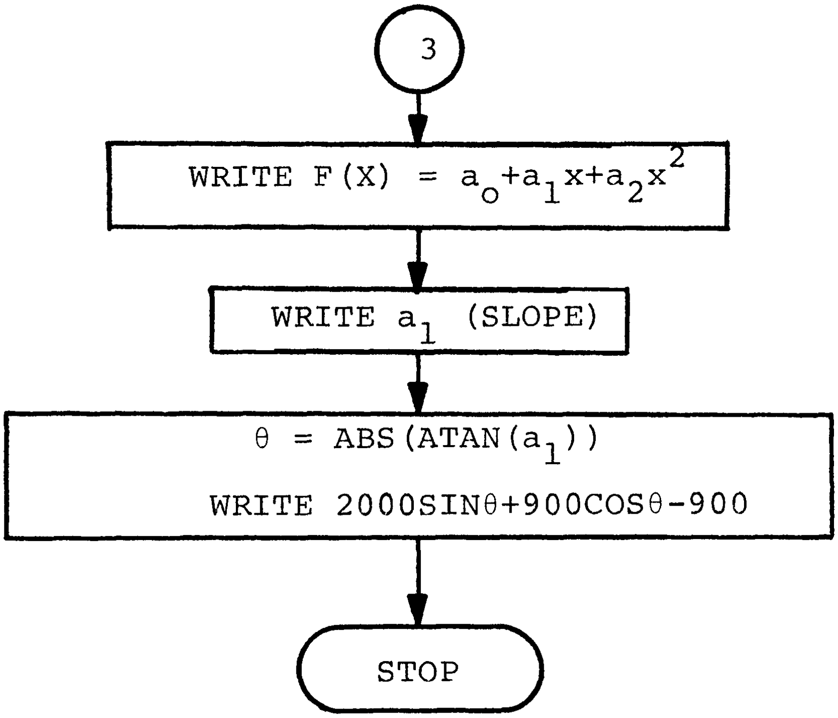
```
C
C
      REAL*4 DUMMY(64,64)
      COMMON/WORKC/DUMMY
      A=0.0
      DO 10 I=1,M
      DO 10 J=1,M
   10 A=A+(DUMMY(I,J))**2
      ROOTSQ=SQRT(A)
      RETURN
      END

/*
//G.FT06F001 DD UNIT=TAPE,VOL=SER=P24013,DISP=(OLD,KEEP),
// DCB=(RECFM=VBS,LRECL=4004,BLKSIZE=4008),LABEL=(2,SL,,IN),
// DSNAME=VIC002
//G.FT05F001 DD UNIT=TAPE,VOL=SER=P24012,DISP=(OLD,KEEP),
// DCB=(RECFM=VBS,LRECL=4004,BLKSIZE=4008),LABEL=(1,SL,,IN),
// DSNAME=ELAINE
//G.FT08F001 DD UNIT=DISK,SPACE=(CYL,(10)),DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=3516,BLKSIZE=3520),DSN=&&TEMP1
//G.FT09F001 DD UNIT=DISK,SPACE=(CYL,(10)),DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=3516,BLKSIZE=3520),DSN=&&TEMP2
//G.DATA DD *
  1
/*
```

APPENDIX B

FLOW CHART

LEAST SQUARES DETERMINATION OF THE ROTATION ANGLE


START
READ
L = ORDER OF POLYNOMIAL FIT
NUMBER = LOWER CORRELATION BOUND
NUMBR 1= UPPER CORRELATION BOUND
I = 1
READ KKK, MMM, K, M, D
KKK = IRST
MMM = ICST
K = IRST+IRMAG-1
M = ICST+IRMAG-1
D = AMAX
1
DATA END
YES
2
NO
D<0.4
NUMBER<M
NUMBR1>M
YES
1
NO
X(I)=M
Y(I)=K
I=I+1
1
2
PRINTER PLOT [Y(I)vsX(I)]
PERFORM LEAST SQUARES ANALYSIS
3


3
WRITE F(X) = $a_o+a_1x+a_2x^2$
WRITE a_1 (SLOPE)
θ = ABS(ATAN(a_1))
WRITE 2000SINθ+900COSθ-900
STOP

```
//OS    JOB CARD
//S1 EXEC RFTGCLG
//C.SOURCE DD *
      REAL*4 X(100),Y(100),A(3,3),X1(6),XY(3),B(3),COEF(3)
      INTEGER*4 R(3),S(3)
C         THIS PROGRAM PERFORMS CURVE SMOOTHING BY POLYNOMIAL LEAST SQUARES.
C         IT WILL NORMALLY BE USED TO FIND THE FIRST OR SECOND ORDER
C         POLYNOMIAL REPRESENTING THE LINE OF CORRELATION MAXIMUMS.
C              THIS PROGRAM IS OUTLINED IN INTRODUCTION TO NUMERICAL
C         METHODS AND FORTRAN PROGRAMMING BY THOMAS RICHARD MC CALLA
C              JOHN WILEY AND SONS, INC. 1967 PAGE 243-245.
      READ(1,100) L,NUMBER,NUMBR1
C        L=THE ORDER OF F(X).  NOTE FOR BEST RESULTS IT SHOULD BE 2.
C        NUMBER= THE LOWER BOUND ON THE RANGE OF GOOD CORRELATION VALUES.
C        NUMBR1= THE UPPER BOUND OF THE RANGE OF GOOD CORRELATION VALUES.
  100 FORMAT(3I5)
      LL=L+1
      I=0
    6 I=I+1
    5 READ(1,110,END=999) KKK,MMM,K,M,D
C              DATA AS OUTPUTED BY THE CORRELATION PROGRAM.
C        KKK= SYSTEM ROW LOCATION
C        MMM=SYSTEM COLUMN LOCATION
C        K=MAXIMUM CORRELATION ROW LOCATION
C        M=MAXIMUM CORRELATION COLUMN LOCATION
  110 FORMAT(4I10,G15.7)
      IF(D.LT.0.4.OR.M.LT.NUMBER.OR.M.GT.NUMBR1) GO TO 5
      WRITE(3,111) K,M,D
  111 FORMAT(' ',2I10,G15.7)
      X(I)=FLOAT(M)
      Y(I)=FLOAT(K)
      GO TO 6
  999 CONTINUE
      I=I-1
      CALL PPLT(X,Y,I)
```

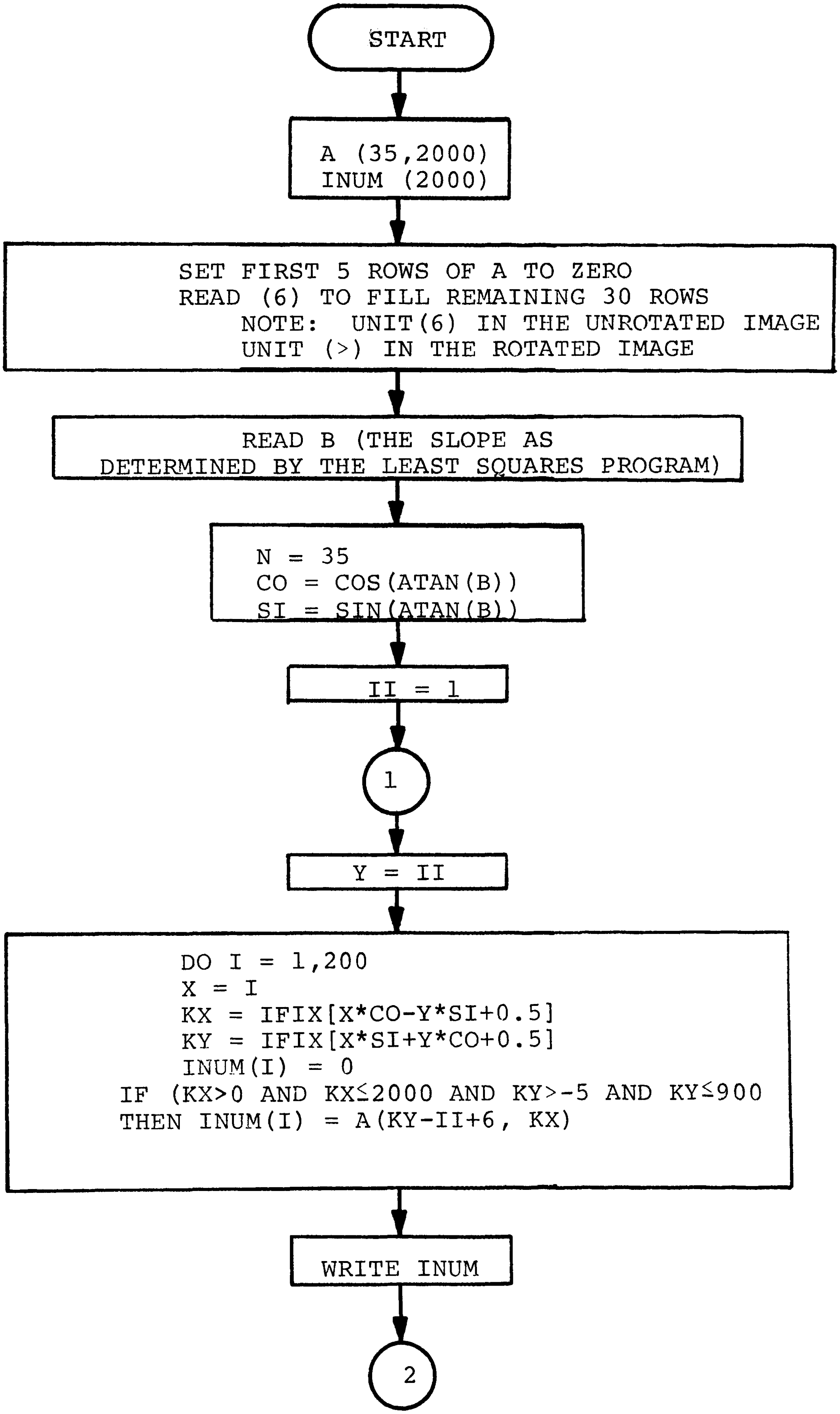
```
      K=2*L
      DO 10 J=1,K
      X1(J)=0.0
      DO 10 M=1,I
   10 X1(J)=X1(J)+X(M)**J
      Y1=0.0
      DO 20 J=1,I
   20 Y1=Y1 + Y(J)
      DO 30 J=1,L
      XY(J)=0.0
      DO 30 K=1,I
   30 XY(J)=XY(J)+Y(K)*X(K)**J
C FORM THE NORMAL MATRIX A AND THE COLUMN MATRIX B
      A(1,1)=FLOAT(I)
      DO 29 K=2,LL
   29 A(1,K)=X1(1+K-2)
      DO 31 K=2,LL
      DO 31 J=1,LL
   31 A(K,J)=X1(K+J-2)
      B(1)=Y1
      DO 33 J=2,LL
   33 B(J)=XY(J-1)
C SOLVE THE EQUATION AX=B  1)INVERT NORMAL MATRIX A,  2) FORM MATRIX PRODUCT FOR
C X,   X=A**(-1)*B.
      CALL MINV(A,LL,DET,R,S)
      WRITE(3,120) DET
  120 FORMAT('0THE DETERMINANT OF A',G15.7)
      CALL GMPRD(A,B,COEF,3,3,1)
      WRITE(3,130) (COEF(J),J=1,LL)
  130 FORMAT('0THE LINEAR LEAST SQUARES SOLUTION OF ALL DATA HAVING CORR
     1ELATION VALUES GREATER THAN 0.4 THAT ALSO BELONG'/' ON THE PAGE AS
     2IS OBVIOUS FROM THE FIRST PROGRAM.'///'0',40X,'F(X)=A0 +A1*X + A2*
     3X**2'//'0',30X,'F(X)=',(G15.7,'  +'))
      BBB=ATAN(COEF(2))
      BBB=ABS(BBB)
```

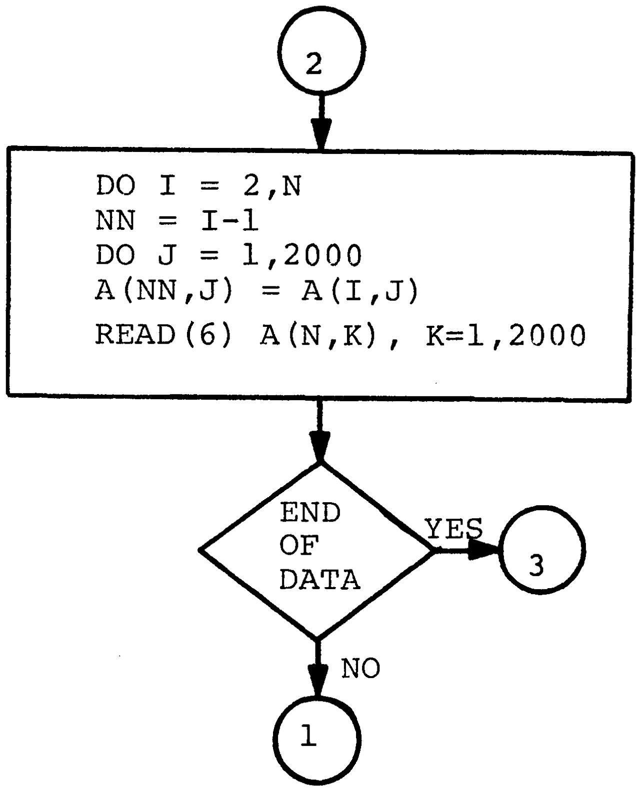
```
      DROW=ABS(2000.0*SIN(BBB)+900.0*COS(BBB)-900.0)
      WRITE(3,140) DROW
  140 FORMAT('0IF THIS NUMBER IS GREATER THAN 40.0, THE ARRAY SIZE IN TH
     1E NEXT PROGRAM MUST BE CHANGED.',G12.4)
      A3=SIN(ATAN(COEF(2)))
      DO 500 J=1,I
  500 Y(J)=Y(J) - X(J)*A3
      CALL PPLT(X,Y,I)
      WRITE(3,333) COEF(2),A3
  333 FORMAT('1','THE SLOPE OF THE CORRELATION MAXIMUM LINE IS',F15.10/'
     10','THE SINE OF THE ANGLE THE SLOPE REPRESENTS IS',F15.10)
      STOP
      DEBUG SUBCHK
      END


/*
//G.DATA DD *
    2    0 1287
THE CORRELATION PROGRAM OUTPUT
/*
```

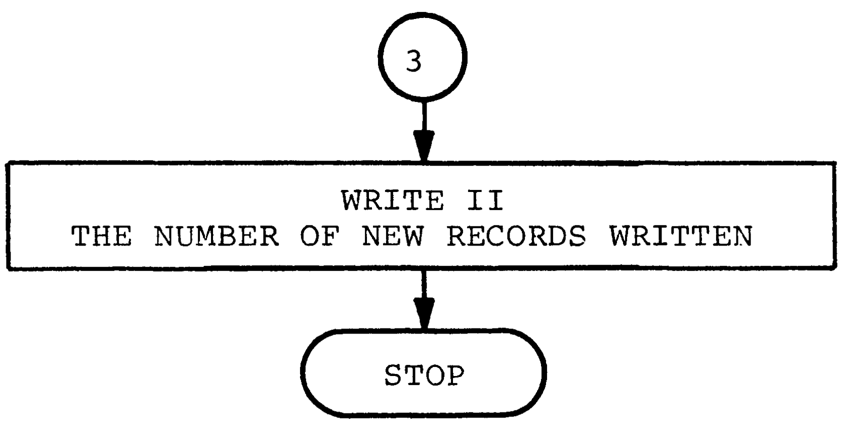
APPENDIX C

FLOW CHART

ROTATION ANGLE


START
A (35,2000)
INUM (2000)
SET FIRST 5 ROWS OF A TO ZERO
READ (6) TO FILL REMAINING 30 ROWS
NOTE: UNIT(6) IN THE UNROTATED IMAGE
UNIT (>) IN THE ROTATED IMAGE
READ B (THE SLOPE AS
DETERMINED BY THE LEAST SQUARES PROGRAM)
N = 35
CO = COS(ATAN(B))
SI = SIN(ATAN(B))
II = 1
1
Y = II
DO I = 1,200
X = I
KX = IFIX[X*CO-Y*SI+0.5]
KY = IFIX[X*SI+Y*CO+0.5]
INUM(I) = 0
IF (KX>0 AND KX≤2000 AND KY>-5 AND KY≤900
THEN INUM(I) = A(KY-II+6, KX)
WRITE INUM
2


2
DO I = 2,N
NN = I-1
DO J = 1,2000
A(NN,J) = A(I,J)
READ(6) A(N,K), K=1,2000
END
OF
DATA
YES
3
NO
1


3
WRITE II
THE NUMBER OF NEW RECORDS WRITTEN
STOP

```
//OS    JOB CARD
//* LIMITS=(T=120,P=100,R=(200))
//S1 EXEC RFTGCLG
//C.SOURCE DD *
C
C ****************  THE ROTATION PROGRAM   *****************
C
      INTEGER*2 A(35,2000),INUM(2000)
      DO 1 I=1,5
      DO 1 J=1,2000
    1 A(I,J)=0
      DO 2 I=6,35
    2 READ(6) (A(I,J),J=1,2000)
C UNROTATED IMAGE IS UNIT 6.  ROATTED IMAGE WILL BE ON UNIT 7.
      READ(1,122) B
  122 FORMAT(F15.7)
      N=35
      CO=COS(ATAN(B))
      SI=SIN(ATAN(B))
      II=1
   15 CONTINUE
      Y=FLOAT(II)
      DO 10 I=1,2000
      X=FLOAT(I)
      KX=IFIX(X*CO-Y*SI+0.5)
      INUM(I)=0
      KY=IFIX(X*SI+Y*CO+0.5)
      IF(KX.GT.0.AND.KX.LE.2000.AND.KY.GT.-5.AND.KY.LE.900) INUM(I)=A(KY
     1-II+6,KX)
   10 CONTINUE
      WRITE(7) INUM
      DO 20 I=2,N
      NN=I-1
      DO 20 J=1,2000
   20 A(NN,J)=A(I,J)
```

```
      READ(6,END=99) (A(N,K),K=1,2000)
      II=II+1
      GO TO 15
   99 CONTINUE
      WRITE(3,101) II
  101 FORMAT('0',I10)
      REWIND 6
      END FILE 7
      REWIND 7
      STOP
      END

/*
//G.FT06F001 DD UNIT=TAPE,VOL=SER=U24005,DISP=(OLD,KEEP),
// DCB=(RECFM=VBS,LRECL=4004,BLKSIZE=4008),LABEL=(1,SL,,IN),
// DSNAME=BILL01
//G.FT07F001 DD UNIT=TAPE,VOL=SER=P24012,DISP=(OLD,KEEP),
// DCB=(RECFM=VBS,LRECL=4004,BLKSIZE=4008),LABEL=(1,SL,,IN),
// DSNAME=ELAINE
//G.DATA DD *
0.0143984556
/*
```

APPENDIX D

MODIFIED CORRELATION PROGRAM

```
//OS   JOB CARD
//* LIMITS=(T=19,P=100,R=(230),C=50)
//S1 EXEC RFTGCLG
//C.SOURCE DD *
C
C
C************   THIS IS THE MODIFIED CORRELATION PROGRAM  **********
C
C
      INTEGER COLOR/6/,INFRA/5/,COLR/8/,IFR/9/,N1(3)/6,6,0/,INVER(16)
      INTEGER KK(100),LL(100)
      INTEGER*2 INUM(2000)
      REAL XKK(100),YLL(100)
      REAL DUMMY(64,64),S(16),CORREL(32,32),RE(100)
      COMPLEX X(64,64,1),Y(64,64,1),R(64,64,1)
      COMMON/WORK/X
      COMMON/WORKC/DUMMY
      COMMON/WORKD/R
      COMMON/WORKE/Y
      COMMON/WORKF/CORREL
  201 FORMAT('0',6E15.3/(' ',6E15.3))
  202 FORMAT('0',6F15.5/(' ',6F15.5))
      M=64
      N=36
      ID=(M-N)/2
  230 FORMAT('1')
    5 READ(1,200,END=999)IRST
  200 FORMAT(I3)
      IW=IRST-1
      INCR=0
      WRITE(3,230)
      IF(IRST-1) 8,8,7
    7 IR=IRST-1
      DO 6 I=1,IR
      READ(COLOR) INUM
```

```
  6 READ(INFRA) INUM
  8 DO 4 I=1,ID
  4 READ(INFRA) INUM
    DO 10 I=1,64
    READ(COLOR) INUM
    WRITE(COLR) INUM
    READ(INFRA) INUM
 10 WRITE(IFR) INUM
    REWIND COLR
    REWIND IFR
    REWIND COLOR
    REWIND INFRA
    DO 99 ICST=1,1925,100
    DO 11 I=1,64
    DO 11 J=1,64
 11 DUMMY(I,J)=0.0
    CALL LOAD(M,COLR,ICST)
    A=AVG(M)
    CALL SSUB(DUMMY,A,DUMMY,M,M,0)
    B=ROOTSQ(M)
    CALL SDIV(DUMMY,B,DUMMY,M,M,0)
    CALL DTOX(M)
    DO 12 I=1,64
    DO 12 J=1,64
 12 DUMMY(I,J)=0.0
    ICS=ICST+ID
    CALL LOAD(N,IFR,ICS)
    A=AVG(N)
    CALL SSUB(DUMMY,A,DUMMY,M,M,0)
    B=ROOTSQ(N)
    CALL SDIV(DUMMY,B,DUMMY,M,M,0)
300 FORMAT(8G10.3)
    CALL DTOY(M,N)
    CALL HARM(X,N1,INVER,S,1,IFER1)
```

```
      CALL HARM(Y,N1,INVER,S,1,IFER2)
      CALL MULTI(M)
   13 CONTINUE
      CALL HARM(R,N1,INVER,S,-1,IFER3)
      CALL MAGI(M)
      CALL CORATE(M,N,ID)
      CALL MAXMIN(ID,IRMAG,ICMAG,ADSMIN,AMAX)
      INCR=INCR + 1
      CMIN=0.0
      IF=4
      RE(INCR)=AMAX
      KK(INCR)=IRMAG + IRST - 1
      LL(INCR)=ICMAG + ICST - 1
      IB=ICST-1
      WRITE(2,310) IW,IB,KK(INCR),LL(INCR),AMAX
      WRITE(3,309) IW,IB,KK(INCR),LL(INCR),AMAX
  309 FORMAT('0',4I10,G15.7)
  310 FORMAT(4I10,G15.7)
   99 CONTINUE
  215 FORMAT('0',2I10,E15.5)
      DO 21 I=1,INCR
      XKK(I)=FLOAT(KK(I))
   21 YLL(I)=FLOAT(LL(I))
      IF(XKK(INCR).EQ.XKK(1).AND.XKK(1).EQ.XKK(INCR-1)) GO TO 308
      CALL PPLT(YLL,XKK,INCR)
  308 CONTINUE
      DO 899 I=1,20
  899 WRITE(2,310)
      GO TO 5
  999 STOP
      DEBUG SUBCHK
      END

      SUBROUTINE MULTI(M)
C
```

```
C
C ******** THIS PROGRAM COMPUTES A RESULTANT ARRAY BY MULTIPLYING
C          THE BASE ARRAY BY THE COMPLEX CONJUGATE OF THE CORRELATING ARRAY
C        THE INPUTS ARE*
C                 M=THE SIZE OF THE BASE MATRIX
C        COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
C
C
      COMPLEX*8 X(64,64,1),Y(64,64,1),R(64,64,1)
      COMPLEX*8 CONJG
      COMMON/WORKE/Y
      COMMON/WORKD/R
      COMMON/WORK/X
      DO 10 I=1,M
      DO 10 J=1,M
   10 R(I,J,1)=X(I,J,1)*CONJG(Y(I,J,1))
      RETURN
      END

      SUBROUTINE MAGI(M)
C
C
C********THIS SUBROUTINE COMPUTES THE MAGNITUDE OF THE COMPLEX M-BY-M ARRAY
C        THE INPUTS ARE*
C                 M=THE SIZE OF THE BASE MATRIX
C        COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
C
C
      COMPLEX*8 R(64,64,1)
      REAL*4 PI(64,64)
      COMMON/WORKD/R
      COMMON/WORKC/PI
      DO 10 I=1,M
      DO 10 J=1,M
   10 PI(I,J)=REAL(R(I,J,1))
```

```
      RETURN
      END
      SUBROUTINE CORATE(M,N,DEL)
C
C
C********THIS SUBROUTINE OBTAINS THE  NONCYCLIC CORRELATION MATRIX FROM
C        CYCLIC M-BY-M INVERSE TRANSFORM.  THE INPUTS ARE*
C                     M=SIZE OF THE BASE MATRIX
C                     N=THE SIZE OF THE CORRELATING MATRIX
C                   DEL=THE SIZE OF THE NONCYCLIC CORRELATING MATRIX.  ITS VALUE
C                       IS 2*(NUMBER OF SHIFTS) + 1.
C        COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
C
C
      INTEGER DELTA,DEL
      REAL*4 PI(64,64),COR(32,32)
      COMMON/WORKC/PI
      COMMON/WORKF/COR
C               TEST COR
      DO 5 I=1,DEL
      DO 5 J=1,DEL
    5 COR(I,J)=0.0
      DELTA=((M-N)/2)-1
      IDELTA=DELTA+2
      DO 10 K=1,IDELTA
      DO 10 L=1,IDELTA
      IR=DELTA+K+1
      IC=DELTA+L+1
   10 COR(IR,IC)=PI(K,L)
      ID=M-IDELTA+2
      DO 20 K=1,IDELTA
      DO 20 L=ID,M
      IR=K+IDELTA-1
      IC=L-ID+1
```

```
   20 COR(IR,IC)=PI(K,L)
      ID=M-DELTA
      DO 30 K=ID,M
      DO 30 L=1,IDELTA
      IR=K-ID+1
      IC=L+DELTA+1
   30 COR(IR,IC)=PI(K,L)
      ID=M-DELTA
      DO 40 K=ID,M
      DO 40 L=ID,M
      IR=K-ID+1
      IC=L-ID+1
   40 COR(IR,IC)=PI(K,L)
      RETURN
      END

      SUBROUTINE MAXMIN(ID,IRMAG,ICMAG,ADSMAX,AMAX)
C ******** THIS SUBROUTINE FINDS THE LOCATION OF THE MAXIMUM WITHIN AN
C            ARRAY OF CORRELATION COEFFICIENTS.
C            INPUTS ARE*
C           ID=THE SIZE OF THE ARRAY OF CORRELATION COEFFICIENTS
C            OUTPUTS ARE*
C              IRMAG= THE ROW LOCATION OF THE MAXIMUM
C              ICMAG=THE COLUMN LOCATION OF THE MAXIMUM
C              ADSMAX= THE EUCLIDEAN DISTANCE TO THE MAXIMUM
C           AMAX=THE MAXIMUM CORRELATION COEFFICIENT
C          COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
      REAL COR(32,32)
      COMMON/WORKF/COR
      IDT=(ID*2)+1
      IRMAG=2
      ICMAG=2
      DO 10 I=1,IDT
      DO 10 J=1,IDT
      IF(COR(I,J).GT.COR(IRMAG,ICMAG)) GO TO 20
```

```
   10 CONTINUE
      GO TO 30
   20 IRMAG=I
      ICMAG=J
      GO TO 10
   30 AMAX=COR(IRMAG,ICMAG)
      IRMAG=IRMAG-ID-1
      ICMAG=ICMAG-ID-1
      A=FLOAT(IRMAG)
      B=FLOAT(ICMAG)
      ADMAX=SQRT(ABS(A)**2 + ABS(B)**2)
      RETURN
      DEBUG SUBCHK
      END

      SUBROUTINE LOAD(M,NAME,ICST)
C *******  THIS SUBROUTINE LOADS AN M BY M ARRAY FROM THE UNIT(NAME)
C          INPUTS ARE*
C            M= THE SIZE OF THE ARRAY TO BE LOADED INTO ACTIVE CORE
C            NAME= THE UNIT (DISK) FROM WHICH THE INFORMATION IS TO BE READ
C            ICST= THE STARTING COLUMN THAT IS TO BE READ
C        COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
      REAL DUMMY(64,64)
      INTEGER*2 INUM(2000)
      COMMON/WORKC/DUMMY
      DO 10 I=1,M
      N=ICST
      READ(NAME) INUM
      DO 20 J=1,M
      MET=INUM(N)
      DUMMY(I,J)=FLOAT(MET)
   20 N=N + 1
   10 CONTINUE
      REWIND NAME
      RETURN
```

```
      END

      SUBROUTINE DTOY(M,N)
C
C
C********THIS SUBROUTINE CHANGES THE N-X-N INPUT ARRAY TO THE COMPLEX
C        M-X-M ARRAY WITH THE REMAINDER OF THE NON N-XN ARRAY FILLED
C        OUT WITH ZEROS.  THE INPUTS ARE*
C                   M=THE SIZE OF THE BASE INPUT ARRAY
C                   N=THE SIZE OF THE CORRELATING ARRAY.
C        COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
C
C
      COMPLEX*8 Y(64,64,1),CMPLX
      REAL*4 DUMMY(64,64)
      COMMON/WORKE/Y
      COMMON/WORKC/DUMMY
      DO 10 I=1,M
      DO 10 J=1,M
   10 Y(I,J,1)=(0.0,0.0)
      IMN=((M-N)/2)
      DO 20 I=1,N
      DO 20 J=1,N
      IC=IMN+J
      IR=IMN+I
   20 Y(IR,IC,1)=CMPLX(DUMMY(I,J),0.0)
      RETURN
      END

      SUBROUTINE DTOX(M)
C
C
C********SUBROUTINE CONVERTS THE BASE INPUT ARRAY INTO COMPLEX FORM.
C        THE INPUTS ARE*
```

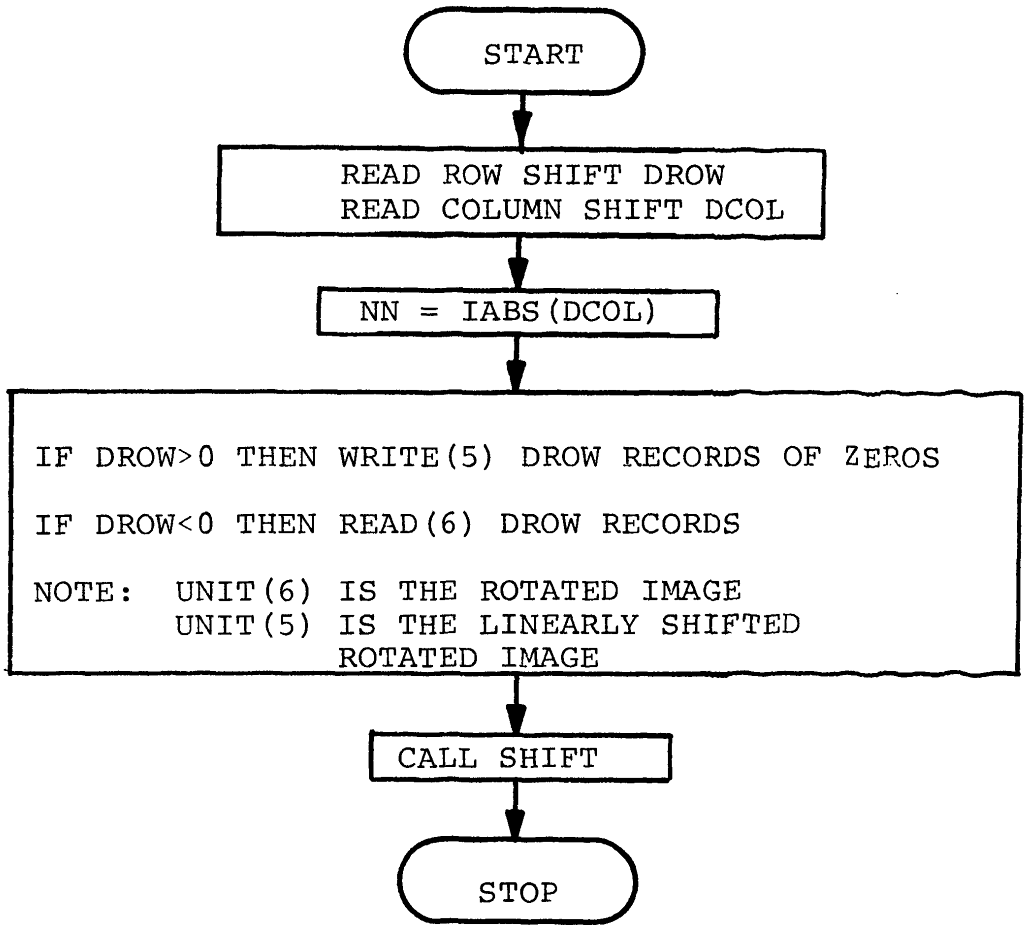
```
C                 M=SIZE OF THE BASE INPUT ARRAY.
C        COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
C
C
      COMPLEX*8 X(64,64,1),CMPLX
      REAL*4 DUMMY(64,64)
      COMMON/WORK/X
      COMMON/WORKC/DUMMY
      DO 10 I=1,M
      DO 10 J=1,M
   10 X(I,J,1)=CMPLX(DUMMY(I,J),0.0)
      RETURN
      END

      FUNCTION AVG(M)
C
C
C********* THIS FUNCTION SUBROUTINE CALCULATES THE STATISTICAL MEAN OF AN
C          M BY M ARRAY.
C          THE INPUTS ARE*
C                 M=SIZE OF THE INPUT ARRAY
C        COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
C
C
      REAL*4 DUMMY(64,64)
      COMMON/WORKC/DUMMY
      A=0.0
      DO 10 I=1,M
      DO 10 J=1,M
   10 A=A+DUMMY(I,J)
      AVG=A/(M*M)
      RETURN
      END

      FUNCTION ROOTSQ(M)
```

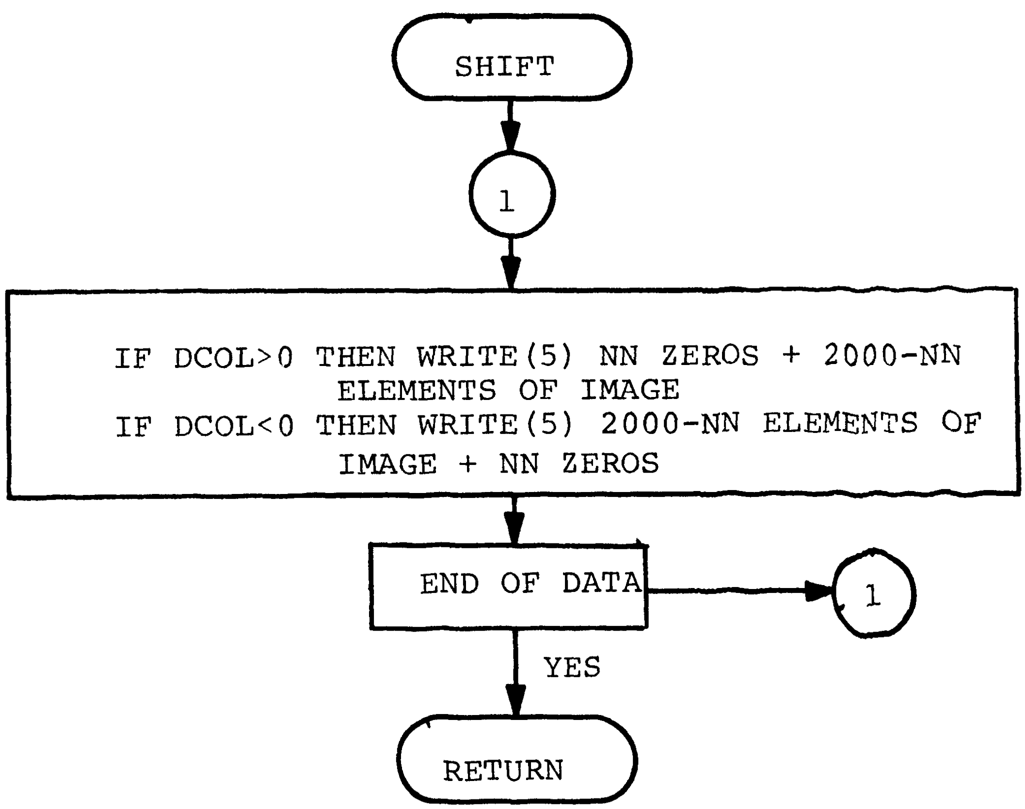
```
C
C
C********THIS FUNCTION SUBROUTINE CALCULATES THE ROOT SUM OF THE SQUARES
C        OF THE SAMPLE SPACE.  INPUTS ARE*
C                 M=SIZE OF THE INPUT ARRAY
C        COMMENT------ALL ARRAYS ARE TRANSFERRED BY COMMON STATEMENTS
C
C
      REAL*4 DUMMY(64,64)
      COMMON/WORKC/DUMMY
      A=0.0
      DO 10 I=1,M
      DO 10 J=1,M
   10 A=A+(DUMMY(I,J))**2
      ROOTSQ=SQRT(A)
      RETURN
      END

/*
//G.FT06F001 DD UNIT=TAPE,VOL=SER=P24013,DISP=(OLD,KEEP),
// DCB=(RECFM=VBS,LRECL=4004,BLKSIZE=4008),LABEL=(2,SL,,IN),
// DSNAME=VIC002
//G.FT05F001 DD UNIT=TAPE,VOL=SER=P24012,DISP=(OLD,KEEP),
// DCB=(RECFM=VBS,LRECL=4004,BLKSIZE=4008),LABEL=(1,SL,,IN),
// DSNAME=ELAINE
//G.FT08F001 DD UNIT=DISK,SPACE=(CYL,(10)),DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=3516,BLKSIZE=3520),DSN=&&TEMP1
//G.FT09F001 DD UNIT=DISK,SPACE=(CYL,(10)),DISP=(,PASS),
// DCB=(RECFM=VBS,LRECL=3516,BLKSIZE=3520),DSN=&&TEMP2
//G.DATA DD *
  1
 80
/*
```

APPENDIX E

FLOW CHART

LINEAR SHIFT PROGRAM


START
READ ROW SHIFT DROW
READ COLUMN SHIFT DCOL
NN = IABS(DCOL)
IF DROW>0 THEN WRITE(5) DROW RECORDS OF ZEROS
IF DROW<0 THEN READ(6) DROW RECORDS
NOTE: UNIT(6) IS THE ROTATED IMAGE
UNIT(5) IS THE LINEARLY SHIFTED ROTATED IMAGE
CALL SHIFT
STOP


SHIFT
1
IF DCOL>0 THEN WRITE(5) NN ZEROS + 2000-NN ELEMENTS OF IMAGE
IF DCOL<0 THEN WRITE(5) 2000-NN ELEMENTS OF IMAGE + NN ZEROS
END OF DATA
1
YES
RETURN

```
//OS    JOB CARD
//* LIMITS=(T=19)
//S1 EXEC RFTGCLG
//C.SOURCE DD *
C
C *************  THE LINEAR SHIFT PROGRAM  ************
C
      INTEGER*2 MO(2000)/2000*0/,NO(30)/30*0/,INUM(2000)
      INTEGER*4 DROW,DCOL
      READ(1,100) DROW,DCOL
  100 FORMAT(2I2)
      NN=IABS(DCOL)
C  THIS PROGRAM WILL READ ON UNIT 6 AND WRITE ON UNIT 5.
      IF(DROW.GT.0) CALL WR(MO,DROW)
      IF(DROW.LT.0) CALL RE(DROW)
      CALL SHIFT(NO,DCOL,NN,DROW)
      STOP
      END

      SUBROUTINE WR(MO,DROW)
      INTEGER*2 MO(2000)
      INTEGER*4 DROW
      DO 10 I=1,DROW
   10 WRITE(5) MO
      RETURN
      END

      SUBROUTINE RE(DROW)
      INTEGER*2 INUM(2000)
      INTEGER*4 DROW
      DROW=-DROW
      DO 10 I=1,DROW
   10 READ(6) INUM
      RETURN
      END
```

```
      SUBROUTINE SHIFT(NO,DCOL,NN,DROW)
      INTEGER*2 NO(NN),INUM(2000)
      INTEGER*4 DROW,DCOL
      I=0
      IF(DCOL.EQ.0) GO TO 20
      J=2000-NN
    5 READ(6,END=99) INUM
      I=I+1
      IF(DCOL.GT.0) WRITE(5) NO,(INUM(K),K=1,J)
      IF(DCOL.LT.0) WRITE(5) (INUM(K),K=1,J),NO
      GO TO 5
   20 CONTINUE
      READ(6,END=99) INUM
      I=I+1
      WRITE(5) INUM
      GO TO 20
   99 I=I+DROW
      WRITE(3,100) I
  100 FORMAT('0THE NUMBER OF NEW FILES WRITTEN=',I7)
      REWIND 6
      END FILE 5
      REWIND 5
      RETURN
      END

/*
//G.FT05F001 DD UNIT=TAPE,VOL=SER=U24005,DISP=(NEW,KEEP),
// DCB=(RECFM=VBS,LRECL=4004,BLKSIZE=4008),LABEL=(2,SL,,IN),
// DSNAME=BILL02
//G.FT06F001 DD UNIT=TAPE,VOL=SER=P24012,DISP=(OLD,KEEP),
// DCB=(RECFM=VBS,LRECL=4004,BLKSIZE=4008),LABEL=(1,SL,,IN),
// DSNAME=ELAINE
//G.DATA DD *
0408
```

APPENDIX F

THE CORRELATION PROGRAM OUTPUT

THE CORRELATION PROGRAM OUTPUT

0	0	3	10	0.4245461
0	20	3	31	0.3405289
0	40	3	51	0.4324251
0	60	3	70	0.4810147
0	80	-1	68	0.3502163
0	100	2	110	0.4261864
0	120	2	130	0.3624469
0	140	2	150	0.2468930
0	160	2	169	0.1874977
0	180	3	189	0.1626363
0	200	12	200	0.2977772
0	220	13	208	0.4321104
0	240	-9	254	0.4229726
0	260	-8	274	0.3935163
0	280	-1	289	0.5291970
0	300	-1	308	0.4894724
0	320	-1	328	0.2569473
0	340	-7	354	0.4414457
0	360	-2	368	0.5174146
0	380	-2	388	0.3841065
0	400	-2	407	0.4355130
0	420	-3	428	0.3554639
0	440	-3	447	0.3637915
0	460	-3	468	0.4838462
0	480	-3	488	0.5441819
0	500	-3	507	0.4603010
0	520	-4	528	0.3038750
0	540	-4	547	0.5374225
0	560	-4	567	0.4329469
0	580	-4	587	0.3472301
0	600	-5	607	0.4690361
0	620	-5	627	0.5892271

0	640	-5	647	0.5394866
0	660	-6	667	0.3144710
0	680	-6	687	0.6412037
0	700	-6	707	0.6016259
0	720	-6	727	0.5401604
0	740	-7	747	0.4538067
0	760	-7	767	0.5269275
0	780	-7	787	0.5467138
0	800	-7	807	0.4141312
0	820	-8	827	0.5458567
0	840	-8	847	0.5912305
0	860	-9	867	0.5005653
0	880	-9	887	0.4800445
0	900	-9	907	0.4194952
0	920	-9	927	0.4182972
0	940	-10	947	0.4895603
0	960	-10	966	0.5062654
0	980	-10	986	0.4219883
0	1000	-11	1007	0.4523775
0	1020	-11	1027	0.4952334
0	1040	-11	1046	0.5576031
0	1060	-12	1067	0.4279774
0	1080	-12	1087	0.4295242
0	1100	-13	1107	0.5110567
0	1120	-13	1127	0.4899008
0	1140	-13	1147	0.5192561
0	1160	-13	1167	0.4675802
0	1180	-13	1187	0.4523173
0	1200	-14	1207	0.4776453
0	1220	-14	1227	0.5021864
0	1240	-14	1247	0.4347105
0	1260	-14	1267	0.4036456
0	1280	-14	1287	0.4463777
0	1300	-14	1307	0.3577109
0	1320	-14	1327	0.3582398

0	1340	-14	1347	0.3007362
0	1360	-14	1368	0.3042935
0	1380	-14	1387	0.2993159
0	1400	-14	1407	0.1610541
0	1420	-14	1428	0.1363699
0	1440	-9	1452	0.2199253
0	1460	3	1472	0.2376674
0	1480	-14	1487	0.1914843
0	1500	7	1506	0.2268670
0	1520	-14	1528	0.3500848
0	1540	-14	1546	0.1605134
0	1560	-14	1561	0.2235008
0	1580	14	1594	0.1651447
0	1600	-14	1611	0.1730883
0	1620	-14	1625	0.2869751
0	1640	-11	1629	0.2159094
0	1660	-13	1670	0.2124974
0	1680	-13	1666	0.1516556
0	1700	13	1694	0.2677257
0	1720	9	1710	0.1143428
0	1740	-5	1728	0.1392764
0	1760	14	1762	0.1132541
0	1780	14	1779	0.8286369E-01
0	1800	-2	1800	0.8028531E-01
0	1820	-6	1807	0.9382695E-01
0	1840	-14	1849	0.1873002
0	1860	2	1870	0.1533663
0	1880	1	1869	0.1986709
0	1900	1	1890	0.1882060
0	1920	-7	1920	0.1415309
0	1940	-14	1932	0.4517217E-01

APPENDIX G

THE RESULTS OF THE MODIFIED CORRELATION PROGRAM AFTER THE IMAGES HAVE BEEN ROTATED

THE RESULTS OF THE MODIFIED CORRELATION PROGRAM AFTER THE IMAGES HAVE BEEN ROTATED.

0	0	4	10	0.4202582
0	100	4	110	0.4233732
0	200	4	209	0.4109524
0	300	4	307	0.4079781
0	400	4	407	0.3939165
0	500	4	507	0.4946582
0	600	4	607	0.7073419
0	700	4	707	0.5194500
0	800	4	807	0.4181349
0	900	4	906	0.5583382
0	1000	4	1007	0.4137938
0	1100	4	1107	0.4967532
0	1200	4	1207	0.5048264
0	1300	4	1307	0.5555022
0	1400	4	1408	0.4993880
0	1500	4	1508	0.5624813
0	1600	4	1608	0.5787229
0	1700	4	1708	0.4767962
0	1800	4	1809	0.6015073
0	1900	4	1910	0.3851098

APPENDIX H

THE CORRELATION OF TWO REGISTERED IMAGES

THE CORRFLATION OF TWO REGISTFRED IMAGES.

0	0	0	2	0.5122082
0	100	1	101	0.4661228
0	200	13	200	0.1864562
0	300	0	300	0.5454082
0	400	0	399	0.4794133
0	500	0	499	0.5331106
0	600	0	599	0.5612405
0	700	0	699	0.5516208
0	800	1	799	0.4769914
0	900	0	898	0.5863765
0	1000	0	999	0.4216608
0	1100	0	1098	0.4715258
0	1200	0	1199	0.4680593
0	1300	0	1299	0.5134293
0	1400	0	1399	0.5182010
0	1500	0	1500	0.5189518
0	1600	0	1600	0.5344371
0	1700	1	1700	0.4406295
0	1800	0	1801	0.6623365
0	1900	0	1902	0.4550398